Bluetooth® Commands

AT Commands
and Application Examples
Reference Guide

MultiTech® Systems

Bluetooth AT Commands
Reference Guide
Products:

Embedded SocketWireless® Bluetooth® Module (MTS2BTSMI)
MultiConnect™ Serial-to-Bluetooth Adapter (MTS2BTA)

PN S000360H, Revision H

Copyright

This publication may not be reproduced, in whole or in part, without prior expressed written permission from Multi-Tech Systems, Inc. All rights reserved. Copyright © 2004-7, by Multi-Tech Systems, Inc.
Multi-Tech Systems, Inc. makes no representations or warranties with respect to the contents hereof and specifically disclaim any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Multi-Tech Systems, Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Multi-Tech Systems, Inc. to notify any person or organization of such revisions or changes.

Revisions

Revision Level	Date	Description
A	08/26/04	Initial release.
B	11/09/04	Updated the product name.
C	04/04/05	Added Bluetooth Adapter to the cover page.
D	07/25/05	Updated commands, **Version 2.8.1.1.0**.
E	01/24/06	Added products list and trademarks/registered trademarks to cover.
F	05/18/07	Updated the Technical Support contact list. Added a note about the PIN: once it is changed, it cannot be obtained or retrieved from the device.
G	08/20/07	Updated commands, **Version 3.6.2.1.0.0** (new feature is Multi-Point connections). Added an Appendix that compares the responses for the two command versions.
H	12/10/07	Changed command examples because the Send commands no longer require a <cr_lf> after the command is typed. Now, the command is executed with <cr> only. Removed term "BlueRadios" from graphics.

Trademarks

SocketWireless, MultiModem and the Multi-Tech logo are registered trademarks of Multi-Tech Systems, Inc.
MultiConnect is a trademark of Multi-Tech Systems, Inc.
Bluetooth is a registered trademark of Bluetooth SIG, Inc.

World Headquarters

Multi-Tech Systems, Inc.
2205 Woodale Drive
Mounds View, Minnesota 55112
Phone: 763-785-3500 or 800-328-9717
Fax: 763-785-9874
Internet Address: http://www.multitech.com

Technical Support

Country	By Email	By Phone
Europe, Middle East, Africa:	support@multitech.co.uk	+44) 118 959 7774
U.S., Canada, all others:	support@multitech.com	800-972-2439 or 763-717-5863

Table of Contents

Chapter 1 – Introduction 5
Overview 5
Changing Configuration 6
Important Notes – Please Read Prior to Continuing 7
Multi-Point (MP) Architecture 8
Multipoint Examples 9
Multi-Point Test Scenarios 10
Repeater Mode 12
Chapter 2 - AT Commands 13
Introduction 13
Attention Command 13
Firmware Version Command 13
Resetting the Device Commands 14
Set/Get Device Information Commands 15
Security Commands 19
Max Number of Bluetooth Connections Commands 19
Set and Read Device Name 20
Set and Read Service Name for Local & Remote Devices 20
Security PIN Settings Commands 22
Class of Device (COD) Commands 23
Write Memory Locations (S-Registers) – Device Configuration Commands 24
Read Memory Locations (S-Registers) Commands 31
Inquiry Commands 31
Connect as Master Commands 32
Set Master Default Bluetooth Address Commands 35
Connect As Slave Command 36
Disconnect Commands 36
About Command/Data Mode 37
About Byte Gaps and Data Latency 37
Data Mode Commands 37
Utilities – Cancel Command 38
Pairing Commands 39
Park and Sniff (Connected Slave) Commands 40
RSSI and Link Quality Commands 42
Audio (SCO) PCM Interface Commands 44
Max Power Command 44
Link Supervisory Timeout Command 45
Variable Storage Commands 45
PIO Logic Commands 46
Transmitting on Specific Channel Only Command 46
Making the Device Discoverable & Undiscoverable Commands 47

Chapter 3 – Using FTP and OPP Modes 48
- **FTP Client** 48
- **FTP Connect/Disconnect Commands** 48
- **FTP Directory Navigation Commands** 49
- **FTP Push/Pull Commands** 51
- **FTP Server** 52
- **OPP Client Commands Introduction** 53
- **OPP Connect/Disconnect Commands** 53
- **OPP Push/Pull Commands** 54
- **OPP Server Commands Introduction** 55
- **Handling OPP Server Business Card Requests Commands** 55

Chapter 3 – Factory Default Settings 58

Chapter 4 – Examples 59
- **Example of a Master Discovery / Connection Sequence** 59
- **Example of a Slave Command Sequence** 60
- **Examples of Server Applications** 60
- **Example Client Applications** 61
- **Multi-Point Example using MTS2BTSMI or MTS2BTA** 62
- **Multi-Point Example Using MTS2BTSMI or MTS2BTA** 64
- **Repeater Example using MTS2BTSMI or MTS2BTA** 66

Appendix A– AT Command Summary Table 68

Appendix B – Responses 72

Appendix C -- UUID Table 79

Appendix D – Class 1 Module Power Consumption 80

Appendix E – Two Versions of Bluetooth Command Responses Comparison 81

Appendix F – Glossary, Acronyms, and Abbreviations 85

Index 86

Chapter 1 – Introduction

Overview

This document describes the protocol used to control and configure the Multi-Tech Systems, Inc. SocketWireless Bluetooth device. The protocol is similar to the industry standard Hayes AT protocol used in telephone modems due to the fact that both types of devices are connection oriented. Appropriate AT commands have been provided to make the module perform the two core actions of a Bluetooth device, which are make/break connections and inquiry. Additional AT commands are also provided to perform ancillary functions.

Just like telephone modems, the serial module powers up into an unconnected state and will respond to inquiry and connection requests. Then, just like controlling a modem, the host or client can issue AT commands, which map to various Bluetooth activities. The command set is extensive enough to allow a host to make connections that are authenticated, encrypted, or not encrypted. The SocketWireless Bluetooth device can be configured, commanded, and controlled through the local serial port or over the Bluetooth RF connection.

Important Notes:

- To provide the best firmware architecture, design, and future profile support, the code is not 100% backward compatible with regard to certain AT commands and responses.
- The CSR (Cambridge Silicon Radio) BC04 chipset in this module is memory-resource limited; therefore, it is recommended that you do NOT fully implement the AT protocol ***similar*** to an AT modem.

Bluetooth Protocol V2.0

For Bluetooth protocol V2.0, the circuit board I/O pin 7 now controls flow control. In V1.2, circuit board I/O pin 3 controlled flow control.

Protocol Change Example:

Previous Protocol V1.2	*Changed to*	*New Protocol V2.0*
ATSW22,3,x,x	changed to	ATSW22,7,x,x
ATSW23,3,x,x	changed to	ATSW23,7,x,x

Comparison of Bluetooth Commands

See Appendix E for a comparison of the responses for Bluetooth AT Commands, Version 3.6.2.1.0.0 with Version 2.8.1.1.0.0.

Changing Configuration

Parameters, such as the Bluetooth Name, Service Name, Class of Device, and Serial Port settings can be viewed and configured. This can be done locally through the serial port UART or from a remote Bluetooth RF link. To configure the SocketWireless Bluetooth device, the device must be in command mode by issuing **+++**. While in command mode, the SocketWireless Bluetooth device will accept ASCII bytes as commands.

You can use the developer board and the RS-232 cable to connect to a PC and pass ASCII characters through the terminal to the SocketWireless Bluetooth device. The communications settings should match the settings used when the SocketWireless Bluetooth device connects. For example, the defaults are:

9600bps
8 bits
No Parity
1 stop bit
Hardware flow control enabled.

Once you change these parameters, you have the option to store them permanently in the non-volatile memory.

Run your favorite terminal emulator, HyperTerminal, or other program.

- Type **AT** on your screen and follow it with a carriage return **<cr>**. You should see "**OK**" returned to you. This will verify that your cable and communications settings are correct. When the SocketWireless device is not connected to another Bluetooth device, you can type the AT commands directly into the SocketWireless' UART; e.g., you do not have to type **+++** to change from data to command mode.
- Now you can enter any of the AT commands discussed in the following sections. Follow these commands by **<cr>**. Valid commands will return an "**OK**" or a valid response. Invalid commands will reply **ERROR**.
- To return to data mode, type **ATMD**. You can now pass or receive data from a remote connected Bluetooth device.

Notes:

- If you change communications parameter settings, remember to change your terminal or emulator communications settings to correspond to the newly created parameter settings.
- AT commands will not echo back to the terminal.

WARNING:

Refrain from streaming ASCII or binary data into the UART when the SocketWireless device does not have a Bluetooth RF connection established while in command mode. This may overrun the UART SocketWireless buffer and will not allow you to make a Bluetooth connection. When the SocketWireless is in the command parser mode, it is looking for valid AT commands followed by **<cr>**.

The command ATSW25 may be used to ignore UART data while unconnected. The DCD will indicate when a connection is made.

Additional Information about the Use of HyperTerminal

Occasionally, we have noticed an unconventional communication operation when using HyperTerminal in conjunction with a PC using the Windows BTW Bluetooth stack and virtual Comm ports. For example, communications may work only in one direction. This will require closing both HyperTerminal programs and starting both HyperTerminal sessions again.

If you experience a Comm error with the BTW Bluetooth stack while connecting to the serial port profile cable icon, select **View** from the Menu Bar and click on **Refresh** to update the service connection. It is recommended to use **Refresh** to update local discovered devices and services to eliminate any "caching" of parameters in memory.

Important Notes – Please Read Prior to Continuing

- Audio is currently supported on Channel 0 only.
- The Bluetooth radio's reset logic is active LOW for BT ver 2.0 modules.
- Unlike the old firmware the new parser design will not accept line feed <lf> after issuing a valid AT Command only a carriage return <cr> shall be used.
- To provide the best firmware architecture, design, and future profile support there is not 100% code backwards compatibility in regards to certain AT Commands and responses. This release firmware is targeted to CSR BC04 platforms. The firmware was developed and tested on BC02 and BC04 platforms. The ATDI command response string no longer returns radio name. There is a separate command to request a remote device name based on CSR's newest design implementation.
- Overall performance improvements from ver 1.2 to ver 2.0:
    - Data throughput for a single point-to-point connection is equal or better than (250Kbps) in fast data mode. While in regular data mode (AT Command parser running) it is only 35Kbps.
    - Current consumption is 15% less.
    - Inquiry responses are much faster.

Multi-Point (MP) Architecture

Using Multi-Point Mode

The ATMP supports any combination of Client/Server connections up to a maximum of 4. The modules are shipped and factory-defaulted as Slaves supporting point-to-point connections. The PIO functionality is the same as previous releases when in this mode. By using the AT Command **ATSSW,3** you can increase the number of Bluetooth connections. However, when you set the module to multiple channels, we automatically reassign the following PIOs:

Channel 00 – PIO(2), Channel 01 – PIO(5), Channel 02 - PIO(7), Channel 03, PIO(8). PIO(5) will no longer toggle at 1Hz.

To maintain backwards compatibility, we maintain the same commands for point-to-point configuration. Commands like **ATRSSI** are the same; but, while in Multi-Point mode, we added the Letter "C" for **ATRSSIC,2** to indicate the remote channel number to request this information from.

When the module is setup for more than one connection, the connections are all in regular data mode. You cannot command the connection(s) in Fast Data Mode because the internal parser needs to interpret where the data is originating from. So, in Multi-Point Mode, there is no such thing as Fast Data Mode. If you are communicating to the ATMP local hardware UART in command mode and an incoming remote connection is established, then the ATMP automatically goes into data mode and is no longer in command mode. You will want to monitor the PIO Connect Channel Status Lines to make sure what state the ATMP is in.

As you are transmitting or receiving data to any given channel, the average inquiry and connection time for each additional connection roughly doubles in time. This occurs because the CPU is busy processing data on each additional channel; this consumes more and more resources with each connection. Therefore, we cannot spec the data throughput or overall system performance since it depends on so many independent and dependant variables related to the number of connections, to the time the connection was established, how much data is being processed, and whether or not the ATMP is being queried by other nearby *Bluetooth* devices. A typical design implementation should be one in which data is not transmitted or received simultaneously from all connections in order to prevent the CPU from crashing.

Note: Multi-Point mode will be overridden by FTP or OPP modes. If the radio makes a client FTP or OPP connection or is configured as an FTP or OPP server, the radio will no longer be in Multi-Point mode.

Receiving Data in Multi-Point Mode

When receiving data in Multi-Point mode, a typical interleaved UART data stream for 4 *Bluetooth* connections coming in on 4 separate channel identifiers will look similar to the following example depending on the order in which they were sent from the remote units to the ATMP:

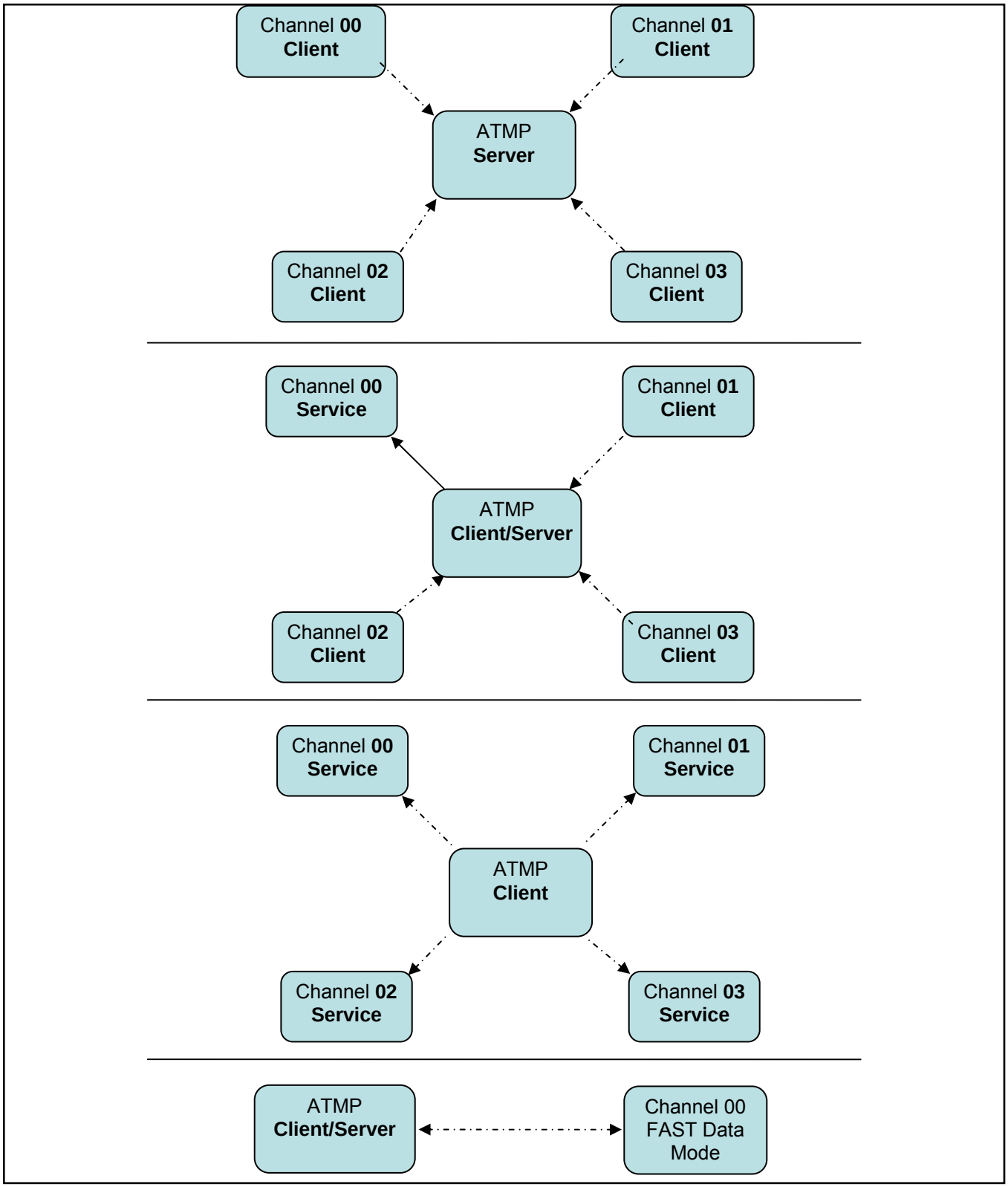
<0x7E>00,data payload1234567890<0x7E>01,data payload12345678901234567890<0x7E>02,data payload123456789012345678901234567890<0x7E>03,datapayload12345678901234567890123456789 01234567890<0x7E>02,data payload123456789012345….

The data payload size is not fixed and is variable in length. The channel header, <0x7E>0#, will be sent whenever data from another channel comes in. The # character represents the channel number of the following data. The header is only sent once if no other channel reports in during the period to lower the communications overhead of the protocol. The ATMP Server needs this method to identify the origination of incoming data for a single physical hardware output UART on the ATMP radio; therefore, data is interleaved as shown above.

Transmitting Data in Multi-Point Mode

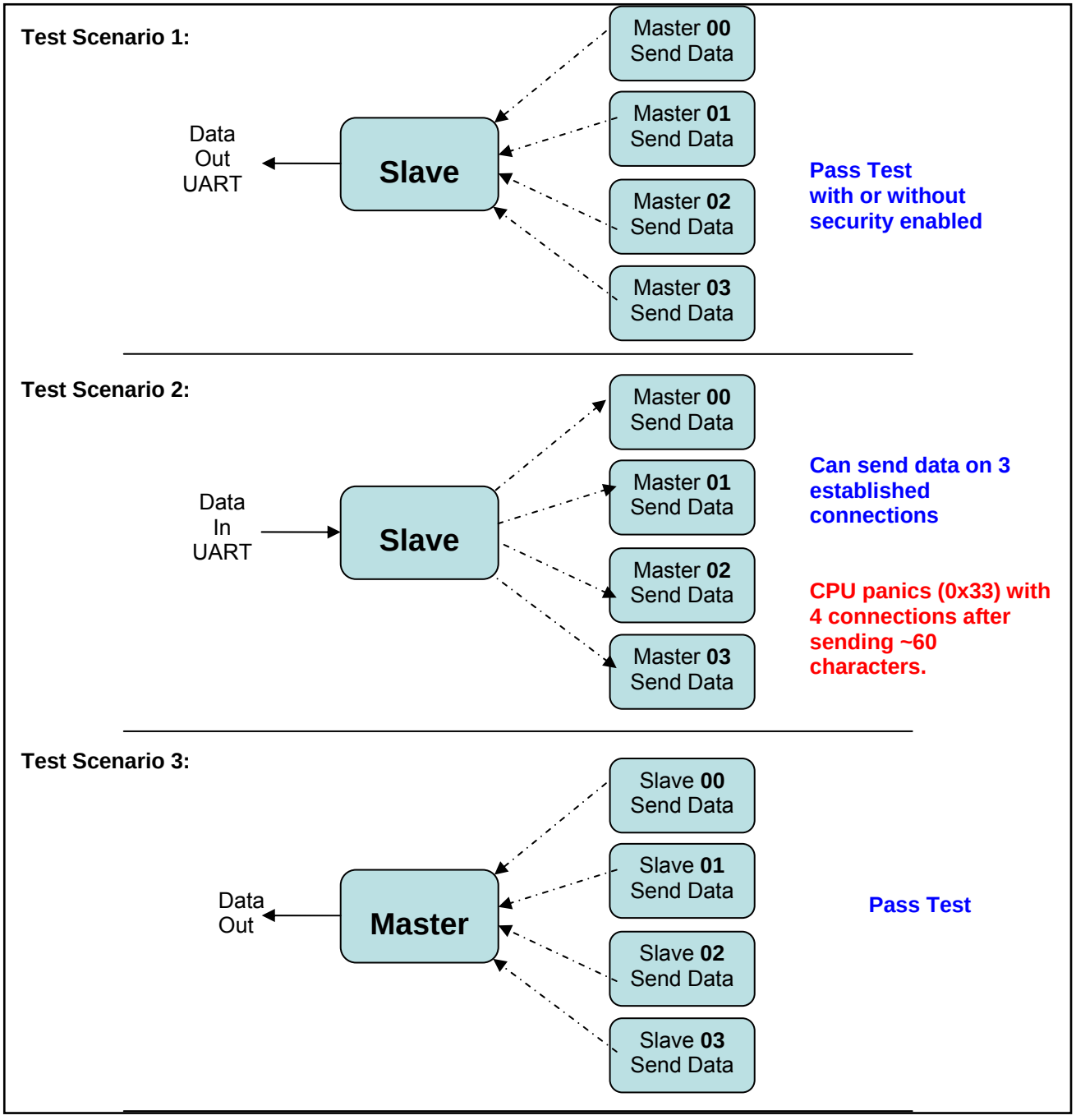
By default, data is broadcast to all connected radios when the local radio is in data mode. The radio can be set to transmit to a specific channel using the **ATSWC** command. (See the Utilities section for more information).

Multipoint Examples

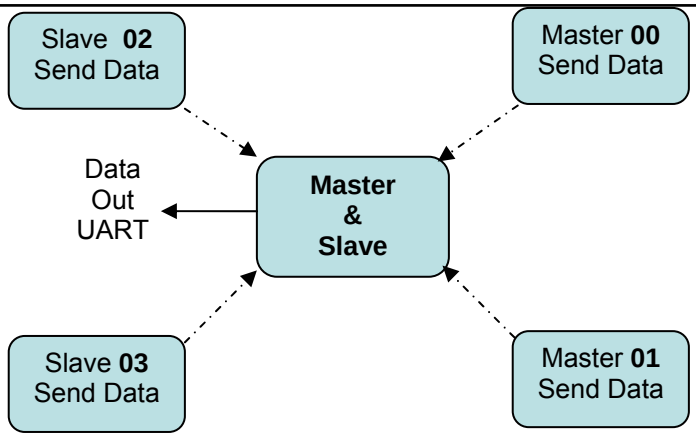


Multi-Point Test Scenarios

The following test examples were all performed by holding down a single key on a PC sending one character every 50-100 msec, which is a worse case to sending more characters less often because the radio has to service interrupts more often. Data was transmitted in one direction only and units under test were all in the same room. Performance will decrease at further distances and if data is full duplex will probably crash with even fewer connections.

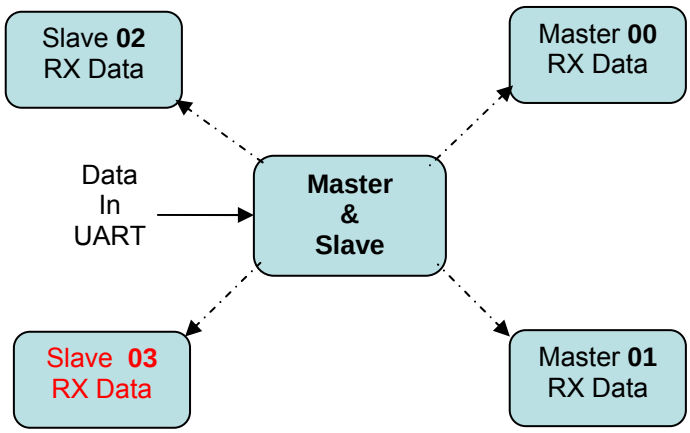


Test Scenario 4:



Pass Test

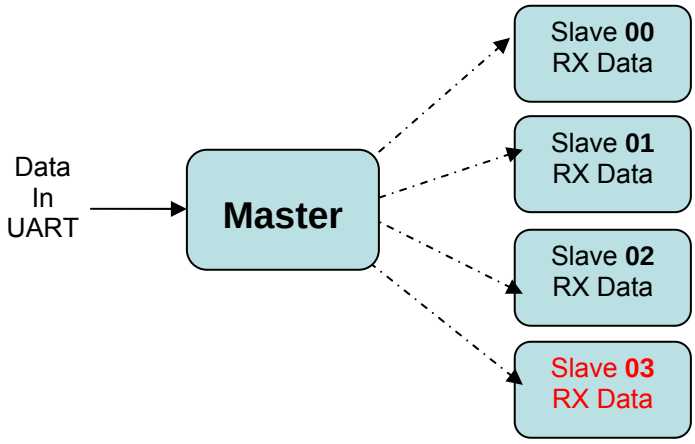
Test Scenario 5:



Can send data on 3 established connections

Unit crashes on 4 connections after sending ~60 characters

Test Scenario 6:

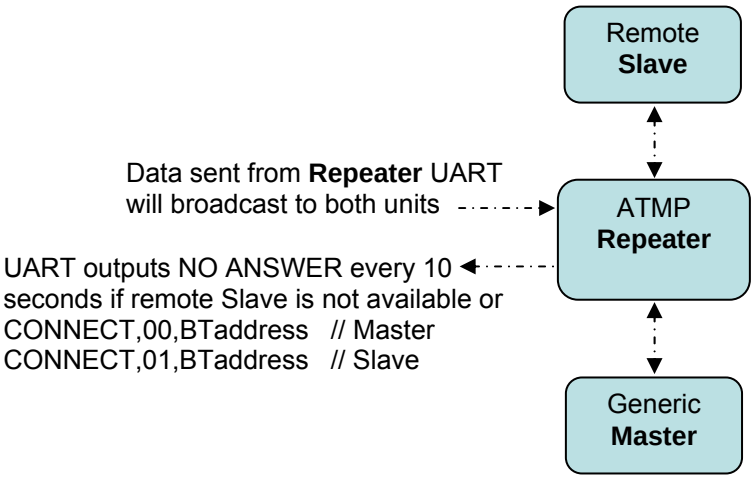


Pass Test for 3 connections

With 4 connections CPU panics (0x33) after ~60 characters

Repeater Mode

Repeater mode can be set using the **ATSW25** and **ATSMA** Commands.



Chapter 2 - AT Commands

Introduction

Command and Response Formats

1. All commands have the following format: **command <cr>**. Where **cr** represents carriage return 0x0D.
2. Valid commands respond with **<cr,lf>OK<cr,lf>** or **<cr,lf>ERROR<cr,lf>**. Where **lf** represents line feed 0x0A. The only exceptions are ATSW20 and ATURST, which do not reply.
3. Allow at least a 100ms delay between subsequent AT commands.
4. HEX vs. Decimal – When writing or entering integer AT command string values, enter them in decimal format. When reading values from memory, they will be returned in hexadecimal format.
5. If using HyperTerminal, the following check box should be disabled. Send line ends with line feeds. If not, the commands will not be submitted correctly.

General Purpose I/O

The send I/O pin for DTR can be configured as input or output. It is referred to in this document as follows:

6 = DTR = PIO (6)

Attention Command

AT – Attention Command Prefix

The prefix **AT** must precede every command except for **+++**. The remainder of the command line contains commands for the SocketWireless. The command script must end with a carriage return.

Example:
Type: AT<cr>
Reply: <cr_lf>OK<cr_lf>

Note:
AT Commands can be in upper or lower case. The only exceptions to this are the SocketWireless Personal Identification alphanumeric Number (PIN), which is case sensitive, and ATOP.

Firmware Version Command

VER Get Firmware Version

Format:
ATVER, ver1

Return Parameters: <Firmware Version>

Example:
Type: **ATVER, ver1**<cr>
Reply: <cr_lf>**OK**<cr_lf>
<cr_lf>**Ver 3.6.2.1.0.0**<cr_lf>

Notes:
- "ver1" is case sensitive; use lower case.
- Make sure this version number matches this document version before proceeding.

Resetting the Device Commands

URST Reset

Tells the SocketWireless device to perform a software reset on the CPU

Format: ATURST

Example:

Type: ATURST<cr>
Reply: None

Notes:

- This unique Command does not reply with “OK” or “ERROR” because of the internal UART data processing limitations and response timing.
- You can send the reset command through the UART or over the Bluetooth RF connection.

FRST Factory Reset

Resets the SocketWireless back to factory defaults. **Example:**

Format: ATFRST

Example:

Type: ATFRST<cr>
Reply: <cr_lf>**OK**<cr_lf>
<cr_lf>**RESET COMPLETE**<cr_lf>
or
<cr_lf>**ERROR**<cr_lf>

Note: You can send the factory reset command through the UART or over the Bluetooth RF connection.

SSW, 0 Set Bypass for Hardware Factory Configuration Reset PIO(4)

Use this command in replace of physically connecting PIO(4) to ground to prevent an inadvertent factory configuration reset.

Format: ATSSW,0,<Enable/Disable>

Parameters: Enable/Disable:
0 = PIO(4) factory reset enabled
1 = PIO(4) factory reset disabled

Example: Type: **ATSSW,0,1**<cr>
Reply: <cr_lf>**OK**<cr_lf>
OR
<cr_lf>**ERROR**<cr_lf>

Read by Using: ATRSW,0

RSW, 0 Read Bypass for Hardware Factory Configuration Reset PIO(4)

Use this command to read the PIO(4) factory reset enable/disable register state.

Format: ATRSW,0

Parameters: **<**Enable/Disable>

Example: Type: **ATRSW,0**<cr>
Reply: <cr_lf>**OK**<cr_lf>
OR
<cr_lf>**00**<cr_lf>

Set by Using: ATSSW,0

Set/Get Device Information Commands

Status Information can be obtained directly from the device. This information is important when managing a connection list of devices in a local area and the current settings of the device.

SI Status Information

Gets specified status information from the LOCAL device.

Format: ATSI,<Status Request>

Parameters: Status Request: Integer 1 to 22
If "Set Using" field is listed, see listed AT command for more information on return parameters.

1 GET BT ADDRESS
Return Parameters: <BT Address>
EXAMPLE:
TYPE: ATSI,1<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>123456789012<cr_lf>

2 GET FRIENDLY NAME
Set Using: ATSN
Return Parameters: <Friendly Name>
EXAMPLE:
TYPE: ATSI,2<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>SocketWireless<cr_lf>

3 GET CURRENT CONNECTION STATUS (CH00-CH03)
Return Parameters: <Connection Status>

Connection Status:
Single Connection Format: (Mode, Ch00 State)
MP Format: (Mode,Ch00 State,Ch01 State,Ch02 State,Ch03 State)
Repeater Format: (Mode,Slave-Ch00 State,Master-Ch01 State)
Modes: 0 = Slave
1 = Auto-Master
2 = Idle
3 = Slave Undiscoverable
5 = Repeater
6 = Mesh
States: 0 = Disconnected
1 = Connected

EXAMPLE:
TYPE: ATSI,3<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>1,0,0,0,0<cr_lf>

Note: *This command does not report the status of any FTP or OPP connections. Use ATSI,22 to obtain FTP/OPP status.*

4 GET SERVICE NAME
Set Using: ATSSN
Return Parameters: <Service Name>
EXAMPLE:
TYPE: ATSI,4<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>COM0<cr_lf>

5 GET CLASS OF DEVICE (COD)

Set Using: ATSC

Return Parameters: <COD>

EXAMPLE:

TYPE: ATSI,5<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>00000000<cr_lf>

6 GET RESPONSE, SECURITY, AUTO SCO, FILTER SETTINGS

Set Using: ATSW24

Return Parameters: <Response Type>,<Security>,<Auto SCO>,<Minor Filter>

EXAMPLE:

TYPE: ATSI,6<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>0,0,0,0<cr_lf>

7 GET CONNECTION, COMM, UNCONNECTED UART, DEFAULT SERVICE MODES

Set Using: ATSW25

Return Parameters: <Power-Up Connection Mode>,<Comm Mode>,
<Unconnected UART Mode>,<Default Service>

EXAMPLE:

TYPE: ATSI,7<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>0,1,0,0<cr_lf>

8 GET UART SETTINGS

Set Using: ATSW20

Return Parameters: <Baudrate>,<Parity>,<Stop Bits> (HEX)

EXAMPLE:

TYPE: ATSI,8<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>0027,0000,0000<cr_lf>

9 GET MASTER AUTO-CONNECT ADDRESS

Set Using: ATSMA

Return Parameters: <BT Address>,<UUID>

EXAMPLE:

TYPE: ATSI,9<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>Not Set!<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>123456789012,1101<cr_lf>

10 GET SLAVE SCAN INTERVALS AND WINDOWS

Set Using: ATSW21

Return Parameters: <psInterval>,<psWindow>,<isInterval>,<isWindow> **(HEX)**

EXAMPLE:

TYPE: ATSI,10<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>0400,0200,0400,0200<cr_lf>

11 GET PIO(5) PULSE RATE

Set Using: ATSW27

Return Parameters: <Pulse Period [ms]> **(HEX)**

EXAMPLE:

TYPE: ATSI,11<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>03E8<cr_lf>

12 GET ESCAPE CHARACTER

Set Using: ATSESC

Return Parameters: <ASCII Char> (HEX)

EXAMPLE:

TYPE: ATSI,12<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>2B<cr_lf>

13 GET INQUIRY AND MASTER CONNECT TIMEOUT SETTINGS

Set Using: ATSW28

Return Parameters: <Inquiry Timeout>,<Master Connect Request Timeout>,<No Data Timeout> **(HEX)**

EXAMPLE:

TYPE: ATSI,13<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>0010,0028,003C<cr_lf>

14 GET MAX TX POWER LEVEL

Set Using: ATSPF

Return Parameters: +/-<Power Level> **(HEX)**

EXAMPLE:

TYPE: ATSI,14<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>default<cr_lf>
OR
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>+0A<cr_lf>

15 GET PIN LOCK MODE

Set Using: ATSW29

Return Parameters: <Lock Mode> **(HEX)**

EXAMPLE:

TYPE: ATSI,15<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>00<cr_lf>

16 GET DEEP SLEEP MODE

Set Using: ATSW30

Return Parameters: <Deep Sleep Mode> (HEX)

EXAMPLE:

TYPE: ATSI,16<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>00<cr_lf>

17 GET SNIFF SETTINGS

Set Using: ATSSNIFF
Return Parameters: <Max Interval>,<MinInterval>,<Attempt>,<Timeout> **(HEX)**
EXAMPLE:

TYPE: ATSI,17<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>Not Set!<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>0000,0000,0000,0000<cr_lf>

18 GET LINK SUPERVISORY TIMEOUT

Set Using: ATLSTO
Return Parameters: <Time> **(HEX)**
EXAMPLE:

TYPE: ATSI,18<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>04<cr_lf>

19 GET LIST OF PAIRED OR SECURED ADDRESSES

Set Using: ATPAIR
Return Parameters: Index,<BT Address> (HEX)
EXAMPLE:

TYPE: *ATSI,19<cr>*
REPLY: *<cr_lf>OK<cr_lf>*
00,<cr_lf>
01,<cr_lf>
02,<cr_lf>
03,<cr_lf>

Note: 00 – 03 are for indexing the stored addresses only. They do not indicate that the address is associated with any specific channel.

20 GET CHANNEL UUIDS

Set Using: ATSSNC
Return Parameters: <Ch00 UUID>,<Ch01 UUID>,<Ch02 UUID>,<Ch03 UUID>
(UUIDs for Ch01-03 will only be returned if enabled using ATSSW3)
EXAMPLE:

TYPE: ATSI,20<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>1101<cr_lf> //Point to point
OR
<cr_lf>1101,1101,1101<cr_lf> //Multipoint with 3 channels

21 GET SPECIFIC TRANSMISSION CHANNEL

Set Using: ATSWC
Return Parameters: <Selected>,<Channel>
EXAMPLE:

TYPE: ATSI,21<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>01,03<cr_lf>

22 GET FTP/OPP CONNECTION STATUS

Return Parameters: <Connection Status>
Connection Status: 0 = Disconnected, 1 = Connected
EXAMPLE:

TYPE: ATSI,22<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>1<cr_lf>

Security Commands

SSW,2 Set Security Level

Sets the Security Level register state.

Format: ATSSW,2,<Security Level>
Parameters: Security Modes:
0 = Link Level – Highest level of security.
1 = Service Level – Provides service information without using PIN.
Factory Default: VM Mode
EXAMPLE:
TYPE: **ATSSW,2,1**<cr>
REPLY: <cr_lf>**OK**<cr_lf>
OR
<cr_lf>**ERROR**<cr_lf>
Read Using: ATRSW,2

RSW,2 Get Security Level

Gets the Security Level register state.

Format: ATRSW,2
Return Parameters: <Boot Mode>
EXAMPLE:
TYPE: **ATRSW,2**<cr>
REPLY: <cr_lf>**OK**<cr_lf>
OR
<cr_lf>**00**<cr_lf>
Set Using: ATSSW,2

Max Number of Bluetooth Connections Commands

SSW, 3 Set Max Connection Number

Sets the maximum number of Bluetooth connections. Enables multi-point mode if number of connections is greater than one.

Format: ATSSW,3,<Number of Connections>
Parameters: Number of Connections: Integer Value 1-4
Factory Default: 1
EXAMPLE:
TYPE: ATSSW,3,1<cr>
REPLY: <cr_lf>OK<cr_lf>

Notes:

- Fast Data Mode is supported with only 1 connection and the ##,00 packet header is never sent. It is recommended you limit the number of connections for your application to maximize performance and security.
- Requires a reset for change to take affect.
- By default data is broadcast to all connected radios when the local radio is in data mode. The radio can be set to transmit to a specific channel using the ATSWC command. (See the Utilities section for more information)

RSW, 3 Read Max Connection Number

Reads the maximum number of Bluetooth connections.

Format: ATRSW,3
Return Parameters: <Max Connections>
EXAMPLE:
TYPE: TRSW,3<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>01<cr_lf>

Set and Read Device Name

SN Set SocketWireless Name

Sets the SocketWireless friendly name (15 alphanumeric characters MAX).

When another Bluetooth device performs a Discovery, this will be the name that is passed to that device. Please take note, unlike the name, the SocketWireless Bluetooth address is fixed (48 bit) at the factory and is unique to every Bluetooth device manufactured.

Format: ATSN,<Name>
Parameters: *Name:* 16 alphanumeric characters MAX
Factory Default: SocketWireless
EXAMPLE:
TYPE: ATSN,MYRADIOS_0123456<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>
Read Using: ATSI,2

RRN Read Remote Radio Name by Bluetooth Address

Gets a remote radio's friendly name using its Bluetooth address.

Format: ATRRN,<BT Address>
Parameters: *BT Address: Bluetooth* Address, 12 hex characters MAX
Return Parameters: <Device Name>
EXAMPLE:
TYPE: ATRRN,0123456789012<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>SocketWireless<cr_lf>
OR
<cr_lf>NO ANSWER<cr_lf>
Note: The timeout for this command is controlled by the master connect timeout in ATSW28.

Set and Read Service Name for Local & Remote Devices

SSN Set Service Name

Set the *Bluetooth* Service Name. Maximum of channel 0 on the local device.

Format: ATSSN,<Service Name>
Parameters: Service Name: 16 alphanumeric characters MAX
Factory Default: "COM0"
Example:
Type: ATSSN,COM0<cr>
Reply: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>
Read Using: ATSI,4 or ATRSN

Note: Requires a reset for change to take effect.

SSNC Set Service Name by Channel

Sets the local *Bluetooth* Service Name and Service UUID by channel number.

Format: ATSSNC,<Channel Number>,<Service Name>,<UUID>

Parameters:

- Channel Number: 0, 1, 2, 3
- Service Name: 16 alphanumeric characters MAX
- UUID: 4 digit, binary profile code (See Appendix C for more info)

Factory Default: "COM0" for Channel 0, "COM1" for Channel 1, "COM2" for Channel 2, "COM3" for Channel 3

EXAMPLE:

TYPE: ATSSNC,0,My Device,1101<cr>

REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Note: Requires a reset for change to take affect.

RSN Read Service Name

Reads the local Bluetooth Service Name.

Format: ATRSN

Return Parameters: <Service Name>

EXAMPLE:

TYPE: ATRSN<cr>

REPLY: <cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>

RSNC Read Service Name by Channel

Reads the local Bluetooth Service Name by channel number.

Format: ATRSNC,<Channel Number>

Parameters: Channel Number: 0-3

Return Parameters: <Service Name>

EXAMPLE:

TYPE: ATRSNC,0<cr>

REPLY: <cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>

RRSN Read Remote Service Name

Reads a remote Bluetooth device Service Name and Service Channel Number for a specific profile.

Format: ATRRSN,<BT Address>,<UUID>

Parameters: BT Address: Bluetooth Address, 12 hex characters MAX
UUID: 4 digit, binary profile code (See Appendix C for more info)

Return Parameters: <Service Channel Number>,<Service Name>

EXAMPLE:

TYPE: ATRRSN,123456789012,1101<cr>

REPLY: <cr_lf>OK<cr_lf>
<cr_lf>01,Bluetooth Serial Port<cr_lf>
<cr_lf>02,Bluetooth Serial Port (2)<cr_lf>
OR
<cr_lf>NO ANSWER<cr_lf>

Note: The timeout for this command is controlled by the master connect timeout in ATSW28.

Security PIN Settings Commands

SSW, 6 Set Pin Request Handling Mode

Sets the PIN (Personal Identification Number) request handling mode. If set to 0 the PIN stored using the ATSP command will automatically be used during PIN exchanges. If set to 1 the prompt "PIN REQUEST" will be output by the radio and the user will need to manually enter a PIN using the ATPR command.

Format: ATSSW,6,<Mode>
Parameters: Mode:
0 = Automatic PIN Request Handling
1 = Manual PIN Request Handling
Factory Default: 0
EXAMPLE:
TYPE: ATSSW,6,1<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>
Read Using: ATRSW,6
Note: Requires a reset for change to take affect.

RSW,6 Read Pin Request Handling Mode

Reads the PIN request handling mode.

Format: ATRSW,6
Return Parameters: <Mode>
EXAMPLE:
TYPE: ATRSW,6<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>00<cr_lf>
Set Using: ATSSW,6

SP Set PIN

Sets the PIN to be used with automatic PIN request handling.

****Warning**** Be careful when entering a new PIN. There is no way to obtain PIN status after it is changed. If the PIN is changed after two units have already been authenticated and connected, you will have to perform a software or hardware reset for the devices to use the new PINs. If this is not done, the two units will still connect using the old stored PIN.

Format: ATSP,<New PIN>,<Old PIN>
Parameters:
New PIN: 16 alphanumeric characters MAX (Caps Sensitive, includes spaces)
Old PIN: 16 alphanumeric characters MAX (Caps Sensitive, includes spaces)
Factory Default: default
EXAMPLE:
TYPE: ATSP,1234567890123456,default<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>
Note: If security is enabled in multipoint mode, all connected slaves will have to use the same PIN. There is no way to assign an individual PIN to each slave.

OP Overwrite PIN

Overwrites the stored PIN without needing the old PIN. To use this command, the PIN must be unlocked using ATSW29.

Format: ATOP,<PIN>
Parameters: PIN: 16 alphanumeric characters MAX (Caps sensitive, includes spaces)
Example:
TYPE: ATOP,1234<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf> //If ATOP has not been enabled with ATSW29.

PR Respond to Manual PIN Request

Allows the user to manually enter a PIN after receiving the PIN REQUEST prompt from the device.

Format: ATPR,<PIN>
Parameters:
PIN: 16 alphanumeric characters MAX (Caps Sensitive, includes spaces)
EXAMPLE:
REPLY: <cr_lf>PIN REQUEST<cr_lf>
TYPE: ATPR,default<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Class of Device (COD) Commands

SC Set Class of Device (COD)

Sets the COD.

Format: ATSC,<COD>
Parameters: A ***Class of Device (COD)*** requires exactly eight 16-bit hex values (0 thru F) based on the *Bluetooth* COD specification names published and maintained by the *Bluetooth* SIG.
Factory Default: 00000000 – Which is undefined since this is set by the user based on the final OEM device it is installed in.
EXAMPLE:
TYPE: ATSC,00020114<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>
Read Using: ATSI,5
Note: Requires a reset for change to take effect.

Write Memory Locations (S-Registers) – Device Configuration Commands

S-Registers refer to memory locations used for configuration. The SW commands are used to assign values to various registers in the device's flash memory that are stored in nonvolatile memory.

SW20 Write UART Settings

Use SW20 to configure UART settings.
The S-Registers refer to memory locations used for configuration.
The S commands are used to assign values to various registers stored in nonvolatile memory.

Format: ATSW20,<Baud Rate>,<Parity>,<Stop Bits>,<Store>

Parameters:

Baud Rates: 1200 – 921.6Kbps, enter ASCII Value from the table below.

Note: Calculating and setting custom baud rates not listed in the table: as long as the equation BAUDRATE *0.004096 produces an integer value, then there will be 0% error in clocking for the baud rate.

Baud Rate	ASCII Value	Error
same	0	-
1200	5	1.73%
2400	10	1.73%
4800	20	1.73%
9600	39	-0.82%
19.2k	79	0.45%
38.4k	157	-0.18%
57.6k	236	0.03%
115.2k	472	0.03%
230.4k	944	0.03%
460.8k	1887	-0.02%
921.6k	3775	0.00%

Parity: 0 = NONE
1 = ODD
2 = EVEN

Stop Bits: 0 = ONE
1 = TWO

Store Parameters: 0 = Do Not Store
1 = Store Parameters in Flash

Factory Default: Baudrate = 39, Parity = 0, Stop Bits = 0 (8, N, 1 w/ hardware flow control RTS/CTS enabled)

EXAMPLE:

TYPE: ATSW20,39,0,0,1<cr> // 9600 8,N,1 store in flash

REPLY: This unique Command does not reply with "OK" or "ERROR" because of internal UART data processing limitations and response timing.

Read Using: ATSI,8

Notes:

- Flow control is always enabled, short CTS/RTS together if not used.
- The RTS line of the radio will be low when the radio is ready to receive data and high when its buffer is full. When RTS goes high wait until it returns to low before sending more data to avoid losing information.
- To reconfigure SocketWireless to the default factory settings, apply 3.3vdc on PIO#4 during initial power up for >2sec.

SW21 Write Slave Scan Intervals and Windows

This command configures the page scan, inquiry scan interval, and window for disconnected slave devices in time slots.

Warning – Setting these will affect the inquiry and connection time. You could inadvertently set the scan interval too long and the window to short on the slave for a master connect request. Unless your application is battery powered slave and power conservation is critical leave at the factory default settings. The minimum Window allowed by the Bluetooth spec is 11.25msec. If you set isWindow = 0 the Slave device will not be discovered by any Master but you can still use the Slaves BT address and connect directly to it from a remote Master.

Format: ATSW21,<psInterval>,<psWindow>,<isInterval>,<isWindow>

Parameters:

- psInterval: Page Scan Interval
  Integer Value 18 to 4096 (11.25ms to 2560ms), 0=Disabled
  Time [ms] = psInterval * 0.625ms
- psWindow: Page Scan Window
  Integer Value 18 to 4096 (11.25ms to 2560ms), 0=Disabled
  Time [ms] = psWindow * 0.625ms
- isInterval: Inquiry Scan Interval
  Integer Value 18 to 4096 (11.25ms to 2560ms), 0=Disabled
  Time [ms] = isInterval * 0.625ms
- isWindow: Inquiry Scan Window
  Integer Value 18 to 4096 (11.25ms to 2560ms), 0=Disabled
  Time [ms] = isWindow * 0.625ms

Factory Default: psInterval = 1024(640ms), psWindow = 512(320ms), isInterval = 1024(640ms), isWindow = 512(320ms)

EXAMPLE:

TYPE: ATSW21,4096,18,4096,18<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,10

Notes:

- Requires a reset for the settings to go into affect.
- ATSW21,4096,18,4096,18 settings will result in a current draw for an unconnected slave of ~1mA average vs. 39mA average at default.
- ATSW21,4096,18,4096,18 settings along with enabling Deep Sleep Mode will result in an unconnected slave current draw of ≈ 350uA average.

SW22 Write PIO Direction

This command configures the direction of the PIOs.

Caution: This has to be done before using PIOs as outputs. Inputting 3.3Vdc into a PIO assigned as output will permanently damage the module.

Format: ATSW22,<PIO#>,<PIO State>,<Store>

Parameters:

- *PIO#:* 3,6,7 (6 & 7 not user definable in MP mode, see notes below)
- *PIO State:*
  0 = Input
  1 = Output
- *Store:*
  0 = Do Not Store
  1 = Store Parameters in Flash

EXAMPLE:

TYPE: ATSW22,6,0,1<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Notes:

Point-to-Point PIO Functionality

- PIO(2) Hard coded as output only. Indicates Bluetooth connection on Ch00.
- PIO(3) Hard coded as input only. High signal of > 1ms will interrupt to wake-up CPU out of deep sleep mode if enabled.
- PIO(4) Hard coded as input only. Triple Purpose PIO. Used for resetting factory defaults on power up and breaking out of Fast Data mode while Bluetooth connected. Also, if strobed, will auto connect to last paired or last connected device.
- PIO(5) Hard coded as output only. Strobes to indicate slave or master inquiry in process.
- PIO(6) User Definable. Defaults as input.
- PIO(7) User Definable. Defaults as input. Can be used for flow control.

Multi-Point PIO Functionality

- PIO(2) Hard coded as output only. Indicates Bluetooth connection on Ch00.
- PIO(3) Hard coded as input only. Defaults as input, high signal of > 1ms will interrupt to wake-up CPU out of deep sleep mode if enabled.
- PIO(4) Hard coded as input only. Triple Purpose PIO. Used for resetting factory defaults on power up and breaking out of Fast Data mode while Bluetooth connected. Also, if strobed, will auto connect to last paired or last connected device.
- PIO(5) Hard coded as output only. Indicates Bluetooth connection on Ch01.
- PIO(6) User definable if no connection on Channel 2. Indicates Bluetooth connection on Ch02.
- PIO(7) User definable if no connection on Channel 3. Indicates Bluetooth connection on Ch03.

SW23 Write PIO Level

This command sets the PIO logic level.

Format: ATSW23,<PIO#>,<Value>,<Store>

Parameters: ATSW23,<PIO#>,<Value>,<Store>

- PIO#: 3,6,7 (6 & 7 not user controllable in MP mode, see notes above in ATSW22)
- PIO State:
  0 = Off (0V)
  1 = On (+V)
- Store:
  0 = Do Not Store
  1 = Store Parameters in Flash

EXAMPLE:

TYPE: ATSW23,6,1,1<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSR21

SW24 Write Response, Security, Auto SCO, Filter Settings

This command configures response, security, auto SCO, and filter settings.

Format: ATSW24,<Response Type>,<Security>,<Auto SCO>,<Minor Filter>

Parameters: >

- *Response Type* (See Appendix B for more information):
  0 = Long Response
  1 = Short Response
  2 = No Verbose Mode (No unsolicited responses will be output)
  - No OKs will come back, but requested information such as ATSI, ATRSW, ATRSN and ATDI responses will.
  - The device will not output CONNECT or DISCONNECT messages.
  - In FTP and OPP modes the radio will still output all necessary messages such as PUSH START and PUSH COMPLETE.
- *Security:* 56-bit encryption is automatically enabled when set to 1.
  UART will reply PAIRED,<BT Address> before the CONNECT,<BT Address> is returned when a connection is made.
  0 = No Authorization
  1 = Authorization Required
  Note: If security is enabled in multipoint mode, all connected slaves will have to use the same PIN. There is no way to assign an individual PIN to each slave.
- *Auto SCO:*
  0 = No Automatic SCO Connect
  1 = SCO Auto Connect Upon Radio Connect.
- *Minor Filter:*
  0 = Disable Minor COD Filter on Inquiry
  1 = Enable Minor COD Filter on Inquiry
  Note: With the minor filter enabled, inquiry results are filtered by the lower 2 bytes of the COD. If the minor filter is disabled all devices are found.

Factory Default: Response Type = 0, Security = 0, Auto SCO = 0, Filter = 0

EXAMPLE:

TYPE: ATSW24,0,0,0,0<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,6

Notes: See Appendix B for differences between short and long response mode. Requires a reset for security to go into affect.

SW25 Write Connection, Comm, Unconnected UART, Default Service Modes

This command configures connection, comm mode, unconnected UART, and default service modes.

Warning – The only way to communicate to the device after setting the radio in "Fast Data Mode" and "ignore UART while unconnected" is to apply 3.3Vdc on PIO(4) during initial power up for >2 sec. These settings are used if you have no control over the source of streaming data into the radio, or you do not know when the radio has made a *Bluetooth* connection, and do not plan on sending any AT commands.

Format: ATSW25,<Power-Up Connection Mode>,<Comm Mode>,<Unconnected UART Mode>, <Default Service Profile>

Parameters:

- *Power-Up Connection Mode:*
  0 = Slave
  1 = Auto-Master (Set ATSMA Command First). Auto-Connect Master always connects using the highest available channel. If none available it will continue to retry.
  2 = Idle Mode. In idle mode the radio is neither slave nor master and draws 1.6mA of current but you can not communicate over the Bluetooth link in this state only through the TX & RX hardware UART.
  3 = Slave Undiscoverable
  5 = Repeater (Set ATSMA Command First). Repeater always uses channel 00 for slave and 01 for Master. Repeater also only uses SPP with no SCO connection.
  6 = Mesh Configuration
  7 = OPP Server (See FTP and OPP Modes Section for more info)
  8 = FTP Server (See FTP and OPP Modes Section for more info)
- *Comm Mode:*
  0 = Fast Data. If data mode is set to "Fast Data Mode" in a Master unit during a manual inquiry "ATDI" and/or connection request the radio connects in slow data mode not fast. This is because you will need the capability to issue commands because in fast data mode the AT command parser is turned off.
  1 = Data
  2 = Command
- *Unconnected UART Mode:*
  0 = Allow Data to Pass While Unconnected
  1 = Ignore Data While Unconnected
- *Default Service Profile:*
  This field no longer has any effect; it can just be set to 0. Service profile UUIDs are now set using the ATSSNC command.

Factory Default: Connection Mode = 0, Comm Mode = 1, Unconnected UART Mode = 0, Default Service Profile = 0

EXAMPLE:

TYPE: ATSW25,0,1,0,0<cr> //Slave radio connects in fast data mode
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,7

Notes:

- Requires a reset for the settings to go into affect.
- The comm mode parameter has no effect when operating in FTP or OPP mode.

SW26 Lock User Settings

This command locks user settings to prevent unauthorized local and remote access..
Warning – This command will lock the PIN.

Format: ATSW26,<PIN>,<Lock/Unlock>

Parameters:

- *PIN:* 16 alphanumeric characters MAX (Caps Sensitive, includes spaces)
- *Lock/Unlock:*
  0 = Unlocked
  1 = Locked

Factory Default: Unlocked

EXAMPLE:

TYPE: ATSW26,default,1<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Note: ATSW26 will still work after locking the user settings, allowing them to be unlocked.

SW27 Write LED Rate

This command sets the LED Pulse Rate on PIO(5).

Format: ATSW27,<Pulse Period>

Parameters:

- *Pulse Rate:* Integer decimal value from 1ms to 60,000ms

Factory Default: 1000

EXAMPLE:

TYPE: ATSW27,1000<cr> //1000 ms
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,11

Note: Used to indicate slave mode operation and inquiry in process. Duty cycle equals 50%.

SW28 Write Inquiry and Master Timeout Settings

This command configures inquiry and master connect timeout settings.

Format: ATSW28,<Inquiry Timeout>,<Master Connect Request Timeout>,<No Data Timeout>

Parameters:

- *Inquiry Timeout:* Integer value from 1 to 40 [seconds]
- *Master Connect Request Timeout:* Integer value from 1 to 40 [seconds]
- *No Data Timeout:* Integer value from 1 to 65535 [seconds]

Factory Default: Inquiry Timeout = 16, Master Connect Request Timeout = 40, No Data Timeout = 60

EXAMPLE:

TYPE: ATSW28,16,40,60<cr> // factory default
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,13

Note: The master connect request timeout also controls the timeouts on the ATRRN, ATRRSN and ATPAIR commands.

SW29 Write PIN Lock Mode

The command configures the PIN lock setting.

Format: ATSW29,<PIN>,<Lock Mode>

Parameters:

- *PIN:* 16 alphanumeric characters MAX (Caps Sensitive, includes spaces)
- *Lock Setting:*
  0 = In Normal operation ATOP is disabled
  1 = Allow ATOP through UART only
  2 = Allow ATOP through UART and over RF Link

Factory Default: 0

EXAMPLE:

TYPE: ATSW29,default,1<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,15

Note: This command enables ATOP, described in the Security PIN Settings section.

SW30 Write Deep Sleep Mode

This command configures the deep sleep mode.

Format: ATSW30,<Deep Sleep Mode>

Parameters:

- *Deep Sleep Mode:*
  0 = Normal Operation never go into deep sleep
  1 = Go into deep sleep whenever possible (While idle, page scan or sniff mode)

Factory Default: 0

EXAMPLE:

TYPE: ATSW30,1<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,16

Notes:

- When the radio is in deep sleep you can not discover or connect to it.
- The UART RX line needs to be pulled high if not active before power is applied to the radio module.
- If there is an active UART RF link the device will need Sniff Mode enabled to allow it to drop into sleep mode when there is no traffic.
- When in deep sleep, the UART will miss the first character while waking up. Send a preamble byte to allow it to wake up and immediately thereafter send the AT Command or data in less than 1 second or the device will go back to deep sleep again. If you are using PIO(3) CPU interrupt a preamble byte is not needed.
- No bytes are lost if sending commands down over the remote RF link side.
- This setting is stored in flash and does not require a reset to take affect. Takes approx. 1 second before the current will drop down to 25-50uA. Allow 5msec. for the CPU unit to come out of deep sleep.
- ATSW21,4096,18,4096,18 settings along with enabling Deep Sleep Mode will result in an unconnected slave current draw of ≈ 350uA average.

Read Memory Locations (S-Registers) Commands

SR21 Read PIO Level

This command reads the PIO logic level.

Format: ATSR21,<PIO#>
Parameters *PIO#:* 2-7
Return Parameters: <Logic Level>
EXAMPLE:
Type*:* ATSR21,3<cr>
Reply: <cr_lf>OK<cr_lf>
<cr_lf>1 <cr_lf>
Set Using: ATSW23

Inquiry Commands

The inquiry command is used to discover all *Bluetooth* radios (within range) that match a certain Class of Device (COD)**.** If the COD is not known it is best to use 00000000 which allows discovery of all devices. You can not be in the default slave mode and perform an inquiry command. Only a Master or a Radio in idle mode can perform an inquiry.

DI Inquire

This command is used to discover other Bluetooth devices.

Format: ATDI,<Max Radios to Discover>,<COD>
Parameters:

- *Max Radios to Discover:* 0-60,000
- *COD:* Exactly 8, 16-bit hex values (0 thru F) based on the Bluetooth COD specification names published and maintained by the Bluetooth SIG. **Note:** With the minor filter enabled using ATSW24, inquiry results are filtered by the lower 4 bytes of the COD. If the minor filter is disabled all devices are found.

Return Parameters: <BT Address>,<COD> (Repeated by number of radios found)

- *An "OK" is returned immediately following this command. "DONE" will appear after all devices have been found, or an inquiry timeout has occurred while searching for the number of devices specified.*

EXAMPLE MASTER:
TYPE: ATDI,1,00000000<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>00A0961F2023,00000000<cr_lf>
<cr_lf>DONE<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Notes:

- Recommend executing an ATUCL command to put the radio in idle mode prior to executing an Inquiry command. See Utilities section.
- Due to a CSR bug, there is no guarantee that the radios returned by an inquiry will all be unique – duplicates can occur. This seems to happen if the inquiry doesn't discover the maximum number of radios. After discovering all of the radios it can, it will then start to return duplicate radios that it has already discovered until the maximum is reached or the inquiry times out.
- The request for friendly name is a separate message request in the new CSR firmware – see ATRRN.

IL Last Inquiry

This command repeats last inquiry.

Format: ATIL

Return Parameters: <BT Address>,<COD>

- An "**OK**" is returned immediately following this command. "**DONE**" will appear after all devices have been found, or an inquiry *timeout* has occurred while searching for the number of devices specified.

EXAMPLE MASTER:

TYPE: ATIL<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>00A0961F2023,00000000<cr_lf>
<cr_lf>DONE<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Note: ATDI command string previously used is stored in flash memory.

Connect as Master Commands

This command is used to connect one module to another. Doing this will enable data transmission bi-directionally. When performing this command, the reply is critical so as to understand where the connection process is. A connection can take several seconds, so when making a connection, if it is not already connected, an **"OK"** will be sent back immediately. Don't mistake this for a connection being complete. A completed connection will return **"CONNECT,00,123456789012"** some time after the command was sent – typically <2 seconds. PIO(2) will go high and stay high or the CD LED on the MTS2BTA will turn on and stay on while a *Bluetooth* connection is established on Channel connection 00.

Note: If either the slave or the master of the connection issues a disconnect, the module will output "DISCONNECT." If a connection is lost unexpectedly, then "LINKLOSS" will be output.

DM Dial as Master

This command creates a connection using the Slave's address and UUID profile code.

Format: ATDM,<BT Address>,<UUID>

Parameters:

- BT Address: Bluetooth Address, 12 hex characters MAX
- UUID: 4 digit, binary profile code (See Appendix C for more info)

Return Parameters: <BT Address> or <Channel Number><BT Address>

Point-to-Point Example:

TYPE: ATDM,123456789012,1101<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>CONNECT,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>PAIRED,123456789012<cr_lf> // if security is enabled
<cr_lf>CONNECT,123456789012<cr_lf>
OR
<cr_lf>NO ANSWER<cr_lf> // if device not present

Multi-Point Example:

TYPE: ATDM,123456789012,1101<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>CONNECT,00,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>PAIRED,123456789012<cr_lf> // if security is enabled
<cr_lf>CONNECT,00,123456789012<cr_lf>
OR
<cr_lf>NO ANSWER<cr_lf> // if device not present

Notes:

- If the remote Slave device is not present or the service is not available, NO ANSWER will reply after the master connect timeout and you will have to try again.
- If security is enabled and the radio is connecting to a PC that has not yet been paired with it, the <cr_lf>PAIRED,12345678912<cr_lf> message may return twice prior to the CONNECT.
- By using the ATDC command connection time will be decreased to approximately 500ms.

DC Dial Channel

This command connects to a remote device by *Bluetooth* address and service channel number (RF Comm ID). This will decrease the connection time to approximately 500ms.

Format: ATDC,<BT Address>,<Service Channel Number>

Parameters:

- BT Address: Bluetooth Address, 12 hex characters MAX
- Remote Service Channel#: 0-3

Return Parameters: <BT Address> or <Channel Number><BT Address>

Point-to-Point Example:

TYPE: ATDC,123456789012,1<cr>

REPLY:
<cr_lf>OK<cr_lf>
<cr_lf>CONNECT,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>PAIRED,123456789012<cr_lf> // if security is enabled
<cr_lf>CONNECT,123456789012<cr_lf>
OR
<cr_lf>NO ANSWER<cr_lf> // if device not present

Multi-Point Example:

TYPE: ATDC,123456789012,1<cr>

REPLY:
<cr_lf>OK<cr_lf>
<cr_lf>CONNECT,00,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>PAIRED, 123456789012<cr_lf> // if security is enabled
<cr_lf>CONNECT,00,123456789012<cr_lf>
OR
<cr_lf>NO ANSWER<cr_lf> // if device not present

Notes:

- Use the ATRRSN command to get a remote service channel number from another device.
- If the remote Slave device is not present or the service is not available, NO ANSWER will reply after the master connect timeout and you will have to try again.
- If security is enabled and the radio is connecting to a PC that has not yet been paired with it, the <cr_lf>PAIRED,12345678912<cr_lf> message may return twice prior to the CONNECT.

DL Dial Last

This command connects to last successful Slave *Bluetooth* address connection over SPP unless ATDM command was executed then the UUID from the ATDM command will be used.

Format: ATDL

Return Parameters: <BT Address> or <Channel Number><BT Address>

Point-to-Point Example:

TYPE: ATDL

REPLY:
<cr_lf>OK<cr_lf>
<cr_lf>CONNECT,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>PAIRED,123456789012<cr_lf> // if security is enabled
<cr_lf>CONNECT,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>NO ANSWER<cr_lf> // if device not present
OR
<cr_lf>ERROR<cr_lf>

Multi-Point Example:

TYPE: ATDL

REPLY:
<cr_lf>OK<cr_lf>
<cr_lf>CONNECT,00,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>PAIRED,123456789012<cr_lf> // if security is enabled
<cr_lf>CONNECT,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>NO ANSWER<cr_lf> // if device not present
OR
<cr_lf>ERROR<cr_lf>

Notes:

- To verify the stored address use the ATLAST command below.
- If the remote Slave device is not present or the service is not available, NO ANSWER will reply after the master connect timeout and you will have to try again.
- If security is enabled and the radio is connecting to a PC that has not yet been paired with it, the <cr_lf>PAIRED,12345678912<cr_lf> message may return twice prior to the CONNECT.

LAST Read Last Address

Gets the last connected Bluetooth device address.

Format: ATLAST

Return Parameters: <BT Address>

EXAMPLE:

TYPE: ATLAST<cr>

REPLY:
<cr_lf>OK<cr_lf>
<cr_lf>000000000000<cr_lf> // Nothing stored
OR
<cr_lf>OK<cr_lf>
<cr_lf>123456789012<cr_lf> // Last connected BT address

Set Master Default Bluetooth Address Commands

SMA Set Master Default Address

This command will set a specific Bluetooth Slave address and service profile into the Master device so on power up the Master will automatically search and connect to a unique Slave device on the highest available channel.

Format: ATSMA,<BT Address>,<UUID>

Parameters:

- *BT Address:* Bluetooth Address, 12 hex characters MAX
- *UUID:* 4 digit, binary profile code (See Appendix C for more info)

EXAMPLE:

TYPE: ATSMA,00A0961F904F,1101<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,9

Notes:

- Execute ATSW25 to set the power up connect mode to auto-master before using the above command if still in slave mode.
- This command is limited to only one connection. Reset module for change to take affect.

MACLR Master Address Clear

This command clears the stored slave address.

Format: ATMACLR

EXAMPLE:

TYPE: ATMACLR<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

SSW,7 Set Auto Connect Master on Data

This command connects automatically to the stored master address when data is present on the UART. The Radio will disconnect when the No Data timeout has been reached, this occurs when no data is being sent to the UART. The ATSMA command must be enabled before this will work. See the ATSW28 command to set the No Data timeout.

Format: ATSSW7,1

EXAMPLE:

TYPE: ATSSW,7,1<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

RSW,7 Read Auto Connect Master on Data

This command reads the AUTO CONNECT MASTER ON DATA enable/disable register state.

Format: ATRSW,7

Return Parameters: <Enable/Disable>

EXAMPLE:

TYPE: ATRSW,7<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>00<cr_lf>

Set Using: ATSSW,7

Connect As Slave Command

Note: If either the slave or the master of the connection issues a disconnect, the module will output "DISCONNECT." If a connection is lost unexpectedly, then "LINKLOSS" will be output.

DS Dial as Slave

This command places the Radio in Slave mode where it waits for a connection to occur from a Master.

Format: ATDS
Return Parameter: OK
EXAMPLE SLAVE:
TYPE: ATDS<cr>
REPLY: <cr_lf>OK<cr_lf>

Disconnect Commands

DH Dial Hang Up

This command will disconnect the current connection on channel 0.

Format: ATDH
EXAMPLE:
TYPE: ATDH<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>DISCONNECT<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Notes:

- If you send this command over the RF link to a remote Multi-Tech Slave, the Slave will disconnect and go into Idle mode not Slave mode by design.
- If a connection should unexpectedly be lost, the response will be *<cr_lf>LINKLOSS<cr_lf>* instead of *<cr_lf>DISCONNECT<cr_lf>*.

DHC Dial Hang Up by Channel

This command will issue disconnect to the specified channel.

Format: ATDHC,<Channel Number>
EXAMPLE:
TYPE: ATDHC,00<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>DISCONNECT,00<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Notes:

- If you send this command over the RF link to a remote Multi-Tech Slave, the Slave will disconnect and go into Idle mode not Slave mode by design.
- If a connection should unexpectedly be lost, the response will be *<cr_lf>LINKLOSS,00<cr_lf>* instead of *<cr_lf>DISCONNECT,00<cr_lf>*.

About Command/Data Mode

1. **Fast Data Mode**
   Fast data mode is currently supported in point to point mode only. There is no such feature in the ATMP Multi-Point code implementation since the parser needs to be on.

   The drawback to this is that once in fast data mode there are limited ways to get out of it like CPU reset, power down, or strobing PIO(4) for >5ms. The advantage of this mode is that the data being sent does not have to be evaluated or processed for AT commands and will allow for a faster effective data throughput of around 250Kbps. To verify you are in fast data mode type +++<cr> it will pass directly through the UART because the AT Command parser is off. Else the +++<cr> will be accepted because the AT Command parser is still on and returns "OK". On the previous 4Mbit legacy modules, the escape characters were fixed and set to "ATMC"; this is no longer implemented.

2. **Command/Data Mode**
   The Command Mode or Data Mode will slow down the throughput when the radio looks for the AT commands. One way to allow minimal overhead transmission burden is to perform all configuration commands, and then place the radio into the fast data mode.

About Byte Gaps and Data Latency

The way Bluetooth is designed and operates random byte gaps of 5 msec to 20 ms are common. Packet size will vary from transmission to transmission. The faster the UART speed the smaller the byte gap delay.

Effective data payload throughput in fast data mode is approximately 250Kbps and 35Kbps in regular data mode when the AT parser looks at each character for ASCII valid command scripts in the data stream. The radio RX has very limited buffering so if you do not use hardware flow control and are transmitting further distances you will quickly overflow the buffer because of RF retransmissions, etc.

When a Bluetooth connection is made the radio modem goes into regular data mode per the power-up factory default settings. This enables the user to remotely configure the radio settings via a remote RF Bluetooth connection. Basically you can setup the radio so no commands are required to be sent from the embedded side of the radios UART. This will prevent any software embedded firmware development or testing for legacy systems.

Data Mode Commands

+++ Put Device into Command Mode

This sequence is used to force the radio into command mode state. If the Radio has been placed in Fast Data Mode this command will have no affect and the typed command will be treated as data. When using this command allow at least **100ms** delay before sending the next AT command.

Format: +++
Return Parameters: If successful an "OK" is returned, or nothing will be returned if already in command mode or fast data mode.

EXAMPLE:

TYPE:	+++<cr>
REPLY:	<cr_lf>OK<cr_lf>

Note: This command is only required if the module is RF CONNECTED, it has no effect if the module is not connected or if it is connected in Fast Data Mode.

SESC Set Command Mode Escape Character

This command sets the radio's escape character, used to put the radio into command mode.

Format: ATSESC,<ASCII Value>
Parameter: *ASCII Value:* Integer value of a non extended ACII character
Factory Default: 43 = "+"
EXAMPLE:

TYPE:	ATSESC,43<cr>
REPLY:	<cr_lf>OK<cr_lf> OR <cr_lf>ERROR<cr_lf>

MD Put Device into Data Mode

This command forces the device into Data Mode. In order for this to have an effect the device has to be CONNECTED.

Format: ATMD

Return Parameters: If successful an "OK" is returned, or nothing will be returned if already in data or fast data mode, and connected. A "NO CARRIER" occurs when the Bluetooth connection has been lost.

EXAMPLE:

TYPE: ATMD<cr>

REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>NO CARRIER<cr_lf>

MF Put Device into Fast Data Mode

This command forces the radio into Fast Data Mode on channel 00 only. In order for this to have an effect, the device has to be CONNECTED. Once in Fast Data Mode, all commands are treated as data. Ways to get out of this mode are to reset power on the radio or strobe PIO(4) while connected.

Format: ATMF

Return Parameters: If successful an "OK" is returned, or nothing will be returned if already in data or fast data mode, and connected. A "NO CARRIER" occurs when the Bluetooth connection has been lost.

EXAMPLE:

TYPE: ATMF<cr>

REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>
OR
<cr_lf>NO CARRIER<cr_lf>

Note: Strobe PIO(4) >5ms to take the device out of fast data into command mode and maintain the Bluetooth RF connection. If not RF connected, the device will be placed in command mode.

Utilities – Cancel Command

UCL Cancel

The UCL command tells the radio to cancel inquiry or connect request commands and then place the radio in Idle Mode. This command can come in handy for a quick exit from commands like inquiry mode if there are no devices in the area and you do not want to wait for an automatic timeout. You can not issue a cancel command while RF connected.

This command can also be used to break out of the auto-master connect mode. If executed while connected in this mode, the radio will disconnect and enter idle mode instead of attempting to connect again.

Format: ATUCL

EXAMPLE:

TYPE: ATUCL<cr>

REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Pairing Commands

PAIR Pair Devices

This command tells the radio in Master mode to pair to a specific *Bluetooth* address.

Format: ATPAIR,<BT Address>

Parameters: *BT Address:* Bluetooth Address, 12 hex characters MAX

EXAMPLE:

TYPE: ATPAIR,00A0961F008F<cr>

REPLY: <cr_lf>OK<cr_lf>
<cr_lf>PAIRED,123456789012<cr_lf>
OR
<cr_lf>ERROR<cr_lf>
OR
<cr_lf>PAIRED,FAILED<cr_lf>

Read Using: ATSI,19

Notes:

- The pairing timeout is 30 seconds.
- The radio needs to be in idle mode prior to pairing.
- Security PINs are exchanged and must be equal for pairing to complete.
- The timeout for this command is controlled by the master connect timeout in ATSW28.

UPAIR Unpair by Index

This command tells the device to unpair from the Bluetooth address stored in index locations 00, 01, 02, and 03.

Format: ATUPAIR,<Index>

Parameters: *Index:* Index location 00, 01, 02 or 03

EXAMPLE:

TYPE: ATUPAIR,00<cr>

REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Notes:

- Use the command ATLAST to view the stored address after pairing.
- Pairing is not the same as a connection, so you will not see inquiry or connection indication. Can be in Master, Slave, or Idle mode to unpair. Will need to reset or cycle power to clear paired address.

UPAIRB Unpair by Bluetooth Address

This command tells the radio to unpair from the specified Bluetooth address.

Format: ATUPAIRB,<BT Address>

Parameters: *BT Address:* Bluetooth Address, 12 hex characters MAX

EXAMPLE:

TYPE: ATUPAIR,123456789012<cr>

REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

CPAIR Clear All Paired Devices

This command tells the radio to unpair from all paired devices.

Format: ATUPAIRC

EXAMPLE:

TYPE: ATCPAIR<cr>

REPLY: <cr_lf>OK<cr_lf>

Park and Sniff (Connected Slave) Commands

Supported in Point-to-Point mode only!

Low Power Modes Using Sniff

Bluetooth connections are master/slave in nature. A master sends packets and a slave has to acknowledge the packet in the next time slot. Time slots in Bluetooth are 625 microseconds wide. This implies that a master will always know when packets will be sent and received, which further means it is able to optimize power usage by switching on power-hungry circuitry only when needed.

A slave, on the other hand, does NOT have prior knowledge of when a packet will be received and has to assume that a packet will be received from a master on every receive slot. This means that it has to leave its receiving circuitry on for most of the receive slot duration. The result of this is high power consumption because a slave with no data transmission still consumes around 5mA.

This problem was identified very early in the evolution of *Bluetooth* (especially since headsets spend all their time as a slave in a *Bluetooth* connection) and it was solved by having a mode called Sniff, with appropriate lower layer negotiating protocol.

Sniff mode during connection is basically an agreement between the slave and its master in which data packets will only be exchanged for N time slots every M slots. The slave can then assume that it will never be contacted during N-M slots, and so can switch its power-hungry circuitry off. The specification goes further by also specifying a third parameter called 'timeout' (T) which specifies 'extra' time slots that the slave will agree to listen for after receiving a valid data packet. Put another way, if a data packet is received by the slave, then it knows that it MUST carry on listening for at least T more slots. If, within that T slot time period, another data packet is received, then the timer is restarted. This mechanism ensures low power consumption when there is no data transfer – at the expense of latency. When there is a lot of data to be transferred, it acts as if sniff mode were not enabled.

It is stated above that during sniff mode, a slave listens for N slots every M slots. The *Bluetooth* specification states that a master can have up to 7 slaves attached to it with all slaves having requested varying sniff parameters. It may, therefore, be impossible to guarantee that each slave gets the M parameter it requested. In light of this, the protocol for enabling sniff mode specifies that a requesting peer specify the M parameter as a minimum and maximum value. This will allow the master to interleave the sniff modes for all slaves attached.

SNIFF Enable Sniff

Manually enables sniff mode for Slave devices connected in time slots.

Format: ATSNIFF,<Max Interval>,<Min Interval>,<Attempt>,<Timeout>

Parameters:

- Max Interval: Integer Value
  Time [ms] = Max Interval * 0.625ms
- Min Interval: Integer Value
  Time [ms] = Min Interval * 0.625ms
- Attempt: Integer Value
  Time [ms] = Attempt * 0.625ms
- Timeout: Integer Value
  Time [ms] = Timeout * 0.625ms

EXAMPLE:

TYPE: ATSNIFF,1600,160,10,160<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>NO CARRIER<cr_lf>

Notes:

- Manually enabling sniff results in an always connected slave using only 2mA average current when no data is sent. It takes about 7 seconds before the current drops after the connection is established. Parameters are lost after the connection is dropped.
- The device will exit sniff mode once the connection is terminated. ATSNIFF must be executed again upon establishing a new connection in order to start sniff mode again.

SSNIFF Enable Auto Sniff

This command stores sniff parameters permanently in flash and auto starts sniff mode after a connection is established.

Warning – Sniff seems to intermittently work using this command. Approximately 1 in 5 connections will not automatically go into sniff mode. Use ATSNIFF to guarantee that sniff mode is started correctly.

Format: ATSSNIFF,<Max Interval>,<Min Interval>,<Attempt>,<Timeout>

Parameters:

- Max Interval: Integer Value
  Time [ms] = Max Interval * 0.625ms
- Min Interval: Integer Value
  Time [ms] = Min Interval * 0.625ms
- Attempt: Integer Value
  Time [ms] = Attempt * 0.625ms
- Timeout: Integer Value
  Time [ms] = Timeout * 0.625ms

EXAMPLE:

TYPE: ATSSNIFF,1600,160,10,160<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,17

CSNIFF Clear Sniff

This command clears sniff parameters.

Format: ATCSNIFF

Example:

Type: ATCSNIFF<cr>
Reply: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

XSNIFF Exit Sniff Mode

This command tells the module to exit and stop "sniffing" RF signals.

Format: ATXSNIFF

Example:

TYPE: ATXSNIFF<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>NO CARRIER<cr_lf>

PARK Enable Park

This command manually enables park mode for the device connected in time slots.

Format: **ATPARK,<MAX Interval>,<MIN Interval>**

Parameters:

- Max Interval: Integer Value
  Time [ms] = Max Interval * 0.625ms
- Min Interval: Integer Value
  Time [ms] = Min Interval * 0.625ms

EXAMPLE:

TYPE: ATPARK,1000,11<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>NO CARRIER<cr_lf>

Notes:

- Allow 7 seconds for the slave to drop its average current draw from 45mA to 3mA and go into PARK. Any UART traffic will cause the Slave radio go back to full operation within 5msec of the first character for 7 seconds before going back into PARK mode without subsequent data. The Slave will remain connected to the Master because the BlueRadios Module only supports one connection.
- Both units are required to support park. Park request is sent from both Master and Slave to go into affect only during an active Bluetooth connection. The order does not matter.

XPARK Exit Park

This command tells the device to exit the park mode.

Format: ATXPARK

EXAMPLE:

TYPE: ATXPARK<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>NO CARRIER<cr_lf>

Note: Unit will immediately exit park.

RSSI and Link Quality Commands

Golden Receive Power Range

The lower threshold level of the Golden Receive Power Range corresponds to a receive power between -56 dBm and 6 dB above the actual sensitivity of the receiver. The upper threshold level is 20 dB above the lower threshold level to an accuracy of +/-6 dB.

RSSI Get RSSI Value

This command is used to obtain the RSSI value for an open connection on channel 0. This is a parameter associated with the ACL connection to a peer device.

Format: ATRSSI

Return Parameters: <RSSI Value>
RSSI Value is in hex, typically from -10 to +31 in integer dB value.

EXAMPLE:

TYPE: ATRSSI<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>+00<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Notes:

- The RSSI value will be +00 if the signal is within the Golden Range.
- The Golden Range min and max value is 1 and 12 respectively for the modules.
- This value is the difference between the measured Received Signal Strength Indication (RSSI) and the limits of the Golden Receive Power Range (see below for definition). Any positive RSSI value returned by the Host side indicates how many dB the RSSI is above the upper limit. Any negative value indicates how many dB the RSSI is below the lower limit. A value of zero indicates that the RSSI is inside the Golden Receive Power Range.
- How accurate the dB values will be depends on the Bluetooth hardware. The only requirements for the hardware are that the Bluetooth device is able to tell whether the RSSI is inside, above, or below the Golden Device Power Range.

RSSIC Get RSSI Value by Channel

This command gets the RSSI Value by channel number.

Format: ATRSSIC,<Channel Number>
Parameters: *Channel Number:* 0, 1, 2, 3
Return Parameters: <RSSI Value>
RSSI Value is in hex, typically from -10 to +31 in integer dB value.
EXAMPLE:
TYPE: ATRSSIC,1
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>+00<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

LQ Get Link Quality

This command gets the link quality of the current connection.

Format: ATLQ
Return Parameters: <Link Quality>
Hex value from 0 to 255 decimal, the measure of Bit Error Rate (BER)
EXAMPLE:
TYPE: ATLQ<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>FF<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Note: Link Quality is a Hex value from 0-255, which represents the quality of the link between two Bluetooth devices. The higher the value, the better the link quality is. Each Bluetooth module vendor will determine how to measure the link quality. In the case for CSR, this value is a measure of BER.

LQC Get Link Quality by Channel

This command gets the Link Quality by channel number.

Format: ATLQC,<Channel Number>
Parameters: *Channel Number:* 0, 1, 2, 3
Return Parameters: <RSSI Value>
Hex value from 0 to 255 decimal, the measure of Bit Error Rate (BER)
EXAMPLE:
TYPE: ATLQ,1<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>FF<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Audio (SCO) PCM Interface Commands

On Voice Version Only

DSCO Dial SCO

This command tells the device to dial and connect the audio channel.

Format: ATDSCO

EXAMPLE:

TYPE: ATDSCO<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>SCO CONNECT<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>SCO FAILED<cr_lf>
OR
<cr_lf>NO CARRIER<cr_lf>

DHSCO Dial Hang Up SCO

This command tells the device to disconnect the audio channel.

Format: ATDHSCO

EXAMPLE:

TYPE: ATDHSCO<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>SCO DISCONNECT<cr_lf>
OR
<cr_lf>NO CARRIER<cr_lf>

Max Power Command

SPF Set Max TX Power Level

Format: ATSPF,<Power Level>,<Sign>

Parameters:

- *Power Level:* Integer from 0 to 10
- *Sign:* + or –

Combined value of level and sign must be in the range of -5 to +10.

Typical *Bluetooth* Industry Values Depending On Device Class Type:
Class 1 = 0dBm to +10dBm

Class 1 Power Table Values:
-5 dBm
0 dBm
5 dBm
0 dBm

Factory Default: 10dBm for Class 1.

EXAMPLE:

TYPE: ATSPF,5,+<cr> // +5dBm
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,14

Note: The default value does not include gains associated with the external antenna (2 dBm). The firmware uses the highest value in the power table that is less than or equal to the requested max transmit power number above.

Link Supervisory Timeout Command

LSTO Link Supervisory Timeout

The command tells the device to drop the connection if the units cannot handshake for x amount of time in seconds.

Format: ATLSTO,<time>
Parameters: *Time:* Integer value from 2 to 41
Factory Default: 4 seconds
Example:

TYPE: ATLSTO,20<cr> // Sets timeout to approx. 20 seconds
REPLY: <cr_lf>OK<cr_lf>
or
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,18
Note:
If timeout is set to a duration that is too short, you may inadvertently drop the connection prematurely if the RF link margin is poor.

Variable Storage Commands

STORE Store Variable

This command allows user to permanently store data – ID's, addresses, etc.

Format: ATSTORE,<Index>,<String>
Parameters: *Index:*
0 = Location 0
1 = Location 1
String:
16 alphanumeric characters MAX
EXAMPLE:

TYPE: ATSTORE,0,1234567890123456<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

READ Read Variable

This command allows user to read a stored variable.

Format: ATREAD,<Index>
Parameters:

Index:
0 = Location 0
1 = Location 1

Return Parameters: <String>
EXAMPLE:

TYPE: ATREAD,0<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>1234567890123456<cr_lf>

Note: Nothing stored will return blank

PIO Logic Commands

SSW,4 Set PIO Logic

This command sets the PIO logic register state. When set to 0, the PIOs (2,5,6 & 7) will be active high. When set to 1, the PIOs will normally be high and then go low when a connection occurs.

Format: ATSSW,4,<PIO Logic>

Parameters:

PIO Logic:
0 = (0 = No Connection, +3.3v = Connection)
1 = (+3.3v = No Connection, 0 = Connection)

Factory Default: 0

EXAMPLE:

TYPE: ATSSW,4,1<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATRSW,4

Note: Requires a reset for change to take affect.

RSW,4 Read PIO Logic

This command sets the PIO logic register state.

Format: ATRSW,4

Return Parameters: <PIO Logic>

EXAMPLE:

TYPE: ATRSW,4<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>00<cr_lf>

Set Using: ATSSW,4

Transmitting on Specific Channel Only Command

SWC Select Transmission Channel

This command allows the user to select a single channel to transmit on in multi-point mode. The device can still receive data from any of the other three channels.

Format: ATSWC,<Select>,<Channel>

Parameters: *Select:*
0 = Deselect the channel as the current transmission channel
1 = Select the channel as the current transmission channel
Channel: 0-3

EXAMPLE:

TYPE: ATSWC,1,0<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Read Using: ATSI,21

Note: If the selected channel is disconnected the radio will default to transmitting on all channels.

Making the Device Discoverable & Undiscoverable Commands

SDIS Set Discoverable

This command makes the device discoverable and connectable.

Format: ATSDIS

EXAMPLE:

TYPE: ATSDIS<cr>
REPLY: <cr_lf>OK<cr_lf>

UDIS Set Undiscoverable

This command makes the radio undiscoverable and unconnectable.

Format: ATUDIS

EXAMPLE:

TYPE: ATSDIS<cr>
REPLY: <cr_lf>OK<cr_lf>

Chapter 3 – Using FTP and OPP Modes

Using the device in FTP and OPP modes is much different from any of the other available connection modes. Connections are not made using ATDM or ATDS and services are not configured using the ATSSNC command. Connection status must be obtained using ATSI22, as ATSI3 contains no information regarding FTP or OPP connections. In addition, only one connection is allowed at a time; so multi-point mode will be disabled once the device is in FTP or OPP mode.

To use the device as an FTP or OPP client, simply use the ATDFTP or ATDOPP command to make a connection to a server. In order to use the device as an FTP or OPP server, the user must set the Power-Up Connection Mode parameter in ATSW25 to either 7 or 8. Following a reset, the device will only have one service exposed, which will either be an FTP or OPP server depending on the value set in ATSW25.

Be aware that due to the extra overhead required to operate in FTP and OPP modes, the data throughput in these modes will be much lower. The device will not operate in Data Mode or Fast Data Mode, so the comm parameter in ATSW25 will have no effect on the throughput. In FTP or OPP client mode, the throughput will be approximately 30kbps if the device is in idle mode prior to making a connection. If the device is not in idle mode, the client throughput will be approximately 15kbps. As an FTP or OPP server, the maximum throughput will be approximately 1.4kbps.

FTP Client

The following FTP (File Transfer Protocol) commands allow the device to be used as an FTP client, enabling it to manipulate files on an FTP server. As the client, the device will initiate the connection with the server; e.g., a PC, PDA, or cell phone. After making a connection the device can browse the server's *Bluetooth* exchange directory, as well as upload and download files to and from the server.

Note: To achieve the maximum data throughput of 30kbps in FTP client mode, execute an ATUCL command prior to making an FTP connection to put the device in idle mode. If the radio is not in idle mode, the data throughput will be approximately 15kbps.

FTP Connect/Disconnect Commands

DFTP Dial FTP

This command creates a connection to the OBEX File Transfer service on a remote device. Once this command has been sent, the device cannot be connected in any other way until the FTP connection is disconnected.

Format: ATDFTP,<BT Address>
Parameters: *BT Address: Bluetooth* Address, 12 hex characters MAX
Return Parameters: <BT Address>
EXAMPLE:

TYPE: ATDFTP,123456789012<cr>
REPLY:
<cr_lf>OK<cr_lf>
<cr_lf>FTP CONNECT,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>PAIRED,123456789012<cr_lf> // if security is enabled
<cr_lf>CONNECT,123456789012<cr_lf>
OR
<cr_lf>FTP NO ANSWER<cr_lf> // if device not present

Notes:

- If the remote slave device is not present or the service is not available, FTP NO ANSWER will reply after the master connect timeout and you will have to try again.
- If security is enabled and the device is connecting to a laptop that has not yet been paired with it, the <cr_lf>PAIRED,12345678912<cr_lf> message may return twice prior to the FTPC CONNECT.
- By default, you will be located in the Bluetooth Exchange Folder of the remote device; this will be the root directory.
- After making an FTP connection, the device will still be in command mode, not data mode.

DHFTP Dial Hang Up FTP

This command will disconnect the current FTP connection.

Format: ATDHFTP

EXAMPLE:

TYPE: ATDHFTP
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>FTP DISCONNECT<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

FTP Directory Navigation Commands

FTPB FTP Browse

This command returns an XML file displaying the contents of the current FTP directory.

Format: ATFTPB

Return Parameters: <XML Directory Information>

EXAMPLE:

TYPE: ATFTPB
REPLY: <cr_lf>OK<cr_lf>
<?xml version="1.0"?><lf>
<!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd"><lf>
<folder-listing version="1.0"><lf>
<folder name="Test Folder" size="0" user-perm="RWD" modified="20060828T203109Z" created="20060828T203018Z" accessed="20060828T203110Z"/><lf>
<file name="TestFile0.txt" size="1" user-perm="RWD" modified="20060828T203058Z" created="20060828T203040Z" accessed="20060828T203058Z"/><lf>
</folder-listing><lf><NULL><NULL>
<cr_lf>BROWSE COMPLETE<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Note: There are two NULL characters at the end of the XML file after the </folder-listing> element.

FTPSUB Move to FTP Sub-Directory

This command moves you to a sub-directory of the current FTP directory.

Format: ATFTPSUB,<Directory>,<Create>

Parameters: *Folder Name:* Path of destination directory, 64 characters MAX
Create:
0 = Do not create new directory if directory does not exist
1 = Create new directory if directory does not exist

EXAMPLES:

TYPE: ATFTPSUB,TestFolder,0<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>
TYPE: ATFTPSUB,TestFolder/NewFolder,1<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Note: The maximum file path length supported by Windows is 255, if this length is exceeded using the ATFTPSUB command you will receive an ERROR message.

FTPUP Move to FTP Parent Directory

This command moves you to the parent directory of the current FTP directory.

Format: ATFTPUP

EXAMPLE:

TYPE: ATFTPUP<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

FTPROOT Move to FTP Root Directory

This command moves you to the FTP root directory.

Format: ATFTPROOT

EXAMPLE:

TYPE: ATFTPROOT<cr>
REPLY: <cr_lf>OK<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Note: The root directory is the Bluetooth Exchange Folder of the remote device.

FTP Push/Pull Commands

FTPPUSH FTP Push

This command transfers a file to the current FTP directory.

Format: ATFTPPUSH,<File Name>,<File Extension>,<File Size>,<Trace>

Parameters:

File Name: 64 characters MAX
File Extension: 4 characters MAX (Ex: .txt,.jpg,.html)
File Size: Size of file in bytes, 4294967295 MAX
Trace:
0 = Do not output transfer progress trace
1 = Output transfer progress trace

Return Parameters: <Bytes Sent>

EXAMPLE WITHOUT TRACE ENABLED:

TYPE: ATFTPPUSH,Test,txt,100,0<cr>
REPLY: <cr_lf>OK<cr_lf>
TYPE: <100 Bytes of Data>
REPLY: <cr_lf>PUSH COMPLETE<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

EXAMPLE WITH TRACE ENABLED:

TYPE: ATFTPPUSH,Test,txt,100,1<cr>
REPLY: <cr_lf>OK<cr_lf>
TYPE: <100 Bytes of Data>
REPLY: <cr_lf>SENT,00000001/00000064<cr_lf>
<cr_lf>SENT,00000051/00000064<cr_lf>
<cr_lf>PUSH COMPLETE<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Notes: If the name of the file matches an existing file in the current FTP directory it will be overwritten without warning.

FTPPULL FTP Pull

This command reads a file from the current FTP directory.

Format: ATFTPPULL,<File Name>,<File Extension>

Parameters:

File Name: 64 characters MAX
File Extension: 4 characters MAX (Ex: .txt,.jpg,.html)

Return Parameters: <File Size>,<File Data>

EXAMPLE:

TYPE: ATFTPPULL,Test,txt<cr>
REPLY: <cr_lf>OK<cr_lf>
cr_lf>File,001E,01234567890123456789012345678 9<cr_lf>
<cr_lf>PULL COMPLETE<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

FTP Server

Setting the Connection Mode in ATSW25 to 8 will cause the device to behave as an FTP Server. As the server, the device will not initiate any connections but will accept one connection at a time from a client. The FTP server functionality of the module is limited: the client can push files to the device, but it cannot browse for or download files.

FTP Server Push Example Using Broadcomm Stack

Enable FTP Server Mode:

Sent: ATSW25,8,1,0,0<cr>
Reply: <cr_lf>OK<cr_lf>
Sent: ATURST<cr>

On a PC, open “Bluetooth Places” and search for devices. After the device is found, double click it and verify its service to be “OBEX FTP on <Friendly Name>”. Copy the file. Then go back to the Bluetooth places window and paste the file on the FTP Service Icon. The PC will then make a connection to the device, push the data, and then disconnect.

Reply: <cr_lf> FTP CONNECT,123456789012<cr_lf>
<cr_lf>PUSH START,New Text File.txt,00000039<cr_lf> //Length is in HEX
This is a test file for FTP server. <cr_lf>
Have a nice day. <cr_lf>
<cr_lf>PUSH COMPLETE<cr_lf>
<cr_lf>FTP DISCONNECT<cr_lf>

OPP Client Commands Introduction

The following OPP (Object Push Profile) commands allow the radio to be used as an OPP client, enabling it to manipulate files on an OPP server. As the client, the device will initiate the connection with the server; e.g., a PC, PDA, or cell phone. After making a connection, the device can push and pull objects.

Note: To achieve the maximum data throughput of 30kbps in OPP client mode, execute an ATUCL command prior to making an OPP connection to put the radio in idle mode. If the radio is not in idle mode, the data throughput will be approximately 15kbps.

OPP Connect/Disconnect Commands

DOPP Dial OPP

This command creates a connection to the OPP service on a remote device. Once this command has been sent, the device cannot make any other connection until the OPP connection is disconnected.

Format: ATDOPP,<BT Address>
Parameters: *BT Address: Bluetooth* Address, 12 hex characters MAX
Return Parameters: <BT Address>
EXAMPLE:

TYPE: ATDOPP,123456789012<cr>
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>OPP CONNECT,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>PAIRED,123456789012<cr_lf> // if security is enabled
<cr_lf>OPP CONNECT,123456789012<cr_lf>
OR
<cr_lf>OK<cr_lf>
<cr_lf>OPP NO ANSWER<cr_lf> // if device not present

Notes:

- *If the remote slave device is not present or the service is not available, OPP NO ANSWER will reply after the master connect timeout and you will have to try again.*
- If security is enabled and the radio is connecting to a laptop that has not yet been paired with it, the <cr_lf>PAIRED,12345678912<cr_lf> message may return twice prior to the OPP CONNECT.
- By default you will be located in the Bluetooth Exchange Folder of the remote device, this will be the root directory.
- After making an OPP connection the radio will still be in command mode, not data mode.

DHOPP Dial Hang Up OPP

This command will disconnect the current OPP connection.

Format: ATDHOPP
EXAMPLE:

TYPE: ATDHOPP
REPLY: <cr_lf>OK<cr_lf>
<cr_lf>OPP DISCONNECT<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

OPP Push/Pull Commands

SBCARD Business Card Push

This command transfers a business card file to the OPP server.

Format: ATRBCARD
Return Parameters: <VCARD Data>
EXAMPLE:

TYPE: ATRBCARD<cr>
REPLY: <cr_lf>OK<cr_lf>
John Smith.vcf,00E3<cr_lf>
BEGIN:VCARD<cr_lf>
VERSION:2.1<cr_lf>
N:Smith;John<cr_lf>
FN:John Smith<cr_lf>
ADR;HOME:;;box ;city;CO;00000;United States of
Ame<cr_lf>
TEL;WORK;VOICE:5551234567<cr_lf>
EMAIL;INTERNET;PREF:johnsmith@domain.com<cr_lf>
X-IRMC-LUID:02000000<cr_lf>
END:VCARD<cr_lf>
<cr_lf>PULL COMPLETE<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

RBCARD Business Card Pull

This command reads a business card file from the OPP server.

Format: ATRBCARD
Return Parameters: <VCARD Data>
EXAMPLE:

TYPE: ATRBCARD<cr>
REPLY: <cr_lf>OK<cr_lf>
John Smith.vcf,00E3<cr_lf>
BEGIN:VCARD<cr_lf>
VERSION:2.1<cr_lf>
N:Smith;John<cr_lf>
FN:John Smith<cr_lf>
ADR;HOME:;;box ;city;CO;00000;United States of
Ame<cr_lf>
TEL;WORK;VOICE:5551234567<cr_lf>
EMAIL;INTERNET;PREF:johnsmith@domain.com<cr_lf>
X-IRMC-LUID:02000000<cr_lf>
END:VCARD<cr_lf>
<cr_lf>PULL COMPLETE<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

OPPPUSH Object Push

This command pushes an object to the OPP server.

Format: ATOPPPUSH,<File Name>,<File Extension>,<Type>,<File Size>

Parameters: *File Name:* 64 characters MAX
File Extension: 3 characters MAX (Ex: .rtf .vcf)
Type: 64 characters MAX
File Size: Size of file in bytes, 65535 MAX

EXAMPLE:

TYPE: ATPIMPUSH,file,vcs,text/x-vcalendar,453<cr>
REPLY: <cr_lf>OK<cr_lf>
TYPE: <100 Bytes of Data>
REPLY: <cr_lf>PUSH COMPLETE<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

OPP Server Commands Introduction

Setting the Connection Mode in ATSW25 to 7 will cause the radio to behave as an OPP Server. As the server the radio will not initiate any connections, but will accept one connection at a time from a client. The OPP server functionality of the module is limited: a client can push all objects to the server, but only business cards can be pulled.

Handling OPP Server Business Card Requests Commands

BCARDR Handle Business Card Request

This command transfers a business card file to an OPP client that has requested a business card. This command can only be used after receiving a PULL REQUEST from the client.

Format: ATBCARDR,<File Name>,<File Extension>,<File Size>

Parameters: *File Name:* 64 characters MAX
File Extension: 3 characters MAX (Ex: .rtf .vcf)
File Size: Size of file in bytes, 65535 MAX

EXAMPLE:

REPLY: <cr_lf>OPP CONNECT,123456789012<cr_lf>
<cr_lf>PULL REQUEST<cr_lf>
TYPE: ATBCARDR,John Smith,vcf,227<cr>
<100 Bytes of Business Card Data>
REPLY: <cr_lf>PULL COMPLETE<cr_lf>
<cr_lf>OPP DISCONNECT<cr_lf>
OR
<cr_lf>ERROR<cr_lf>

Note: Most devices such as PC's and phones will rename the business card file to <Contact's Full Name>.<File Extension> once it has been received.

OPP Server Send Business Card Example Using Broadcom Stack

Enable OPP Server Mode:

Sent: ATSW25,7,1,0,0<cr>
Reply: <cr_lf>OK<cr_lf>
Sent: ATURST<cr>

On a PC, open "Bluetooth Places" and search for devices. After the device is found, double click it and verify its service to be "OBEX Object Push on <Friendly Name>". Right click the icon and select Send Business Card. When prompted, select the vCard file you would like to send. The PC will connect to the server and push the business card, disconnecting when complete.

Reply: <cr_lf> OPP CONNECT,123456789012<cr_lf>
<cr_lf>PUSH START,John Smith.vcf,00E3<cr_lf> //Length is in HEX
BEGIN:VCARD<cr_lf>
VERSION:2.1<cr_lf>
N:Smith;John<cr_lf>
FN:John Smith<cr_lf>
ADR;HOME:;;box ;city;CO;00000;United States of Ame<cr_lf>
TEL;WORK;VOICE:5551234567<cr_lf>
EMAIL;INTERNET;PREF:johnsmith@domain.com<cr_lf>
X-IRMC-LUID:02000000<cr_lf>
END:VCARD<cr_lf>
<cr_lf>PUSH COMPLETE<cr_lf>
<cr_lf>OPP DISCONNECT<cr_lf>

OPP Server Receive Business Card Example Using Broadcom Stack

Enable OPP Server Mode:

Sent: ATSW25,7,1,0,0<cr>
Reply: <cr_lf>OK<cr_lf>
Sent: ATURST<cr>

On a PC, open "Bluetooth Places" and search for devices. After the device is found, double click it and verify its service to be "OBEX Object Push on <Friendly Name>". Right click the icon and select Receive Business Card. The PC will make a connection to the server, and the radio will output "PULL REQUEST." Then use the ATBCARDR command to send a business card to the PC.

Reply: <cr_lf> OPP CONNECT,123456789012<cr_lf>
<cr_lf>PULL REQUEST<cr_lf>
Sent: <cr>ATBCARDR, John Smith,vcf,227<cr>
<227 Bytes of vCard Data>
Reply: <cr_lf>PULL COMPLETE<cr_lf>
<cr_lf>OPP DISCONNECT<cr_lf>

OPP Server Send Data Example Using TransSend

Enable OPP Server mode and set the COD of the device to a known COD, such as that of a laptop so that TransSend will discover the radio.

Sent: ATSW25,7,1,0,0<cr>
Reply: <cr_lf>OK<cr_lf>
Sent: ATSC,0072010c<cr> // TransSend uses COD filtering, so use a known COD.
Reply: <cr_lf>OK<cr_lf>
Sent: ATURST<cr>

Open your browser and find the picture or block of text you would like to send. After you have done this, highlight all you wish to send. Then right click it and go to “TransSend to device with Bluetooth”. Click note or picture. Wait for TransSend to search for devices and select the desired device. If sending text, select “Plain Text,” to send the data click OK.

Reply: <cr_lf> OPP CONNECT,123456789012<cr_lf>
<cr_lf>PUSH START,John Smith.vcf,0015<cr_lf> //Length is in HEX
This is a test note.<cr_lf>
<cr_lf>PUSH COMPLETE<cr_lf>
<cr_lf>OPP DISCONNECT<cr_lf>

Chapter 3 – Factory Default Settings

Note: Appling 3.3Vdc on PIO(4) for >2 sec. during initial power up will revert all user definable settings to the factory defaults shown below. Note: This does not apply to the MultiConnect Adapter.
The other option for resetting factory defaults is to use the software command ATFRST. The only exception for these two options is the name of the device (friendly name) and the PIN – these will not reset if you had changed them already. Allow approximately 5 seconds for the device to reconfigure.

- Bypass Hardware Factory Reset = Enabled
- Escape Character = '+'
- Max Connection Number = 1
- Radio Name = "SocketWireless"
- Country Code = North America and Europe
- Boot Mode = Virtual Machine (VM) Mode
- Service Name = "COM0" (Ch0), "COM1" (Ch1), "COM2" (Ch2), "COM3" (Ch3)
- PIN = "default"
- COD = 00000000
- UART Setting = 9600 Baud, 8 Data Bits, No Parity, 1 Stop Bit (8,N,1)
- Hardware flow control RTS/CTS = Enabled
- Page Scan Interval = 1024 (640ms)
- Page Scan Window = 512 (320ms)
- Inquiry Scan Interval = 1024 (640ms)
- Inquiry Scan Window = 512 (320ms)
- PIO Directions (Point to Point) = 6-In, 7-In
- PIO Directions (Multipoint) = 6-Out, 7-Out
- Response Type = Long Response
- Security = Disabled (If the security flag is enabled a factory reset of parameters does not disable security)
- Security Level = Link Level, if security flag is enabled services are provided only with PIN exchange.
- Automatic SCO Connect = Disabled
- Minor Filter = Disabled
- Default Boot Mode = Slave
- Radio Status = 1,0 (Slave Disconnected)
- Default Comm Mode = Data
- Unconnected UART Mode = Allow Data to Pass While Unconnected
- *Bluetooth* Service Profile = Serial Port Profile (SPP)
- Lock User Settings = Disabled
- PIO(5) LED Rate = 1000ms
- Inquiry Timeout = 16s
- Master Connect Request Timeout = 40s
- PIN Lock Mode = ATOP Disable
- Deep Sleep Mode = Never Go Into Deep Sleep
- Pairing Timeout = 30 seconds
- Class1 Radio Max Transmit Power = 15dbm. Class2 module will still have a max of 4dbm output performance even though you can set it to 15.
- Link Supervisory Timeout = ~4s

Miscellaneous Note:

Retries for over-the-air RF-guaranteed data packet is set for ***indefinite***.

Chapter 4 – Examples

Example of a Master Discovery / Connection Sequence

From Power Up and No Connection:

1. Perform an Inquiry to obtain **BT Address** (unless it is already known).
   Sent : ATUCL**<cr>** // Clears radio state and places in Idle Mode
   Reply: <cr_lf>OK**<cr_lf>**
   Sent : ATDI,1,00000000 {Class of Device}**<cr>** // Looks for only one Bluetooth device
   Reply: <cr_lf>00A0961F2023,00000104,SocketWireless**<cr_lf>**
   <cr_lf>DONE**<cr_lf>**

2. Perform a Master Connect over SPP using the **BT Address.**
   Sent : ATDM, 00A0961F2023,1101**<cr>** // SPP connection
   Reply: <cr_lf>CONNECT**,**00A0961F008F **<cr_lf>** // Returns Slave BT address radios is in Data Mode

3. Place radio into Fast Data Mode.
   Sent : ATMF**<cr>** // Places radio in Fast Data Mode
   Reply:<cr_lf>OK**<cr_lf>**

4. Send Data.

Note: When sending commands from the Slave when the Slave connects in Fast Data Mode (ATSW25/or issuing ATMF). All valid AT commands are sent through the Slaves UART will be interpreted and responded by the Master radio as if it was the local Slave radio. Basically in this configuration from the Slave end you can obtain status and configure from the remote Master radio. This is a unique feature that may be useful in some applications but can confuse the user if you think you are talking to the local Slave UART.

Get Out of Data Mode and Check Status:

1. Delay at least 50 milliseconds; this could be less or more.

2. Get into Command Mode.
   Sent : +++**<cr>** // Default escape sequence of characters
   Reply:<cr_lf>OK**<cr_lf>**

3. Check Status
   Sent : AT**<cr>**
   Reply:<cr_lf>OK**<cr_lf>**

4. Or send any AT Command example:
   Sent : ATSI,0**<cr>**
   Reply:<cr_lf>MULTITECH**<cr_lf>**

Example of a Slave Command Sequence

From Power Up

1. Perform an inquiry and search for Slave Bluetooth device with a PC or other embedded unit.
2. Send a connection request from PC or embedded device to the Slave.
3. Wait for a connection
   Reply: <cr_lf>CONNECT,{SLAVE ADDRESS}<cr_lf> // SPP Connected
4. Send Data.

Note:
This command sequence assumes the SocketWireless is in factory default in which it automatically comes up and is connectable as a Slave from a Master request.

Get Out of Data Mode and Check Status

1. Delay at least 50 milliseconds.
2. Check Status, perform a Disconnect …etc.
   Sent : AT**<cr>**
   Reply: <cr_lf>OK**<cr_lf>**

Examples of Server Applications

Headset Point-to-Point

Sent: ATSW25,0,1,0,0<cr> // Sets connect mode to slave
Reply: <cr_lf>OK<cr_lf>
Sent:ATSSNC,0,Headset0,1108<cr> // Sets service and name
Reply: <cr_lf>OK<cr_lf>
Sent: ATSC,00200404<cr> // Sets COD to a known headset COD
Reply: <cr_lf>OK<cr_lf>
Sent: ATSW25,2,0,0,0<cr> // Sets no response mode
Reply: <cr_lf>OK<cr_lf>

Headset Multipoint

Sent: ATSSW,3,4<cr > // Sets connection number to 4
Reply: <cr_lf>OK<cr_lf>
Sent: ATSSNC,0,Headset0,1108<cr > // Sets service and name
Reply: <cr_lf>OK<cr_lf>
Sent: ATSC,00200404<cr> // Sets COD to a known headset COD
Reply: <cr_lf>OK<cr_lf>
Sent: ATSW25,2,0,0,0<cr> // Sets no response mode
Reply: <cr_lf>OK<cr_lf>

DUN Point-to-Point

Sent: ATSSNC,0,dun0,1103<cr> // Sets service and name
Reply: <cr_lf>OK<cr_lf>
Sent: ATSW25,0,1,0,0<cr> // Sets connect mode to Slave
Reply: <cr_lf>OK<cr_lf>
Sent: ATSC,00420210<cr> // Sets COD to a known DUN COD
Reply: <cr_lf>OK<cr_lf>

DUN Multi-Point

Sent: ATSSW,3,4<cr> // Sets connection number to 4
Reply: <cr_lf>OK<cr_lf>
Sent: ATSSNC,0,dun0,1103<cr> // Sets service and name
Reply: <cr_lf>OK<cr_lf>
Sent: ATSC,00420210<cr> // Sets COD to a known DUN COD
Reply: <cr_lf>OK<cr_lf>

LAN Point-to-Point

Sent: ATSSNC,0,lan0,1102<cr> // Sets service and name
Reply: <cr_lf>OK<cr_lf>
Sent: ATSW25,0,1,0,0<cr> // Sets connect mode to slave
Reply: <cr_lf>OK<cr_lf>
Sent: ATSC,00020300<cr> // Sets COD to a known LAN COD
Reply: <cr_lf>OK<cr_lf>

LAN Multipoint

Sent: ATSSW,3,4<cr > // Sets connection number to 4
Reply: <cr_lf>OK<cr_lf>
Sent: ATSSNC,0,lan0,1102<cr> // Sets service and name
Reply: <cr_lf>OK<cr_lf>
Sent: ATSC,00020300<cr> // Sets COD to a known LAN COD
Reply: <cr_lf>OK<cr_lf>

Example Client Applications

DUN

Only supports the connection.

Sent: ATSC,00420210<cr> // Sets COD to a known DUN COD
Reply: <cr_lf>OK<cr_lf>
Sent: ATDM, <DUN Address>,1103<cr> // Sets auto connect address
Reply: <cr_lf>OK<cr_lf>

Multi-Point Example using MTS2BTSMI or MTS2BTA

One Master and Four Slave Devices

Note: Attaching the fourth slave device can overload the connections and cause all connections to be terminated. Increasing the distance between Bluetooth devices will effect performance and could further limit the number of connections.

Setup hardware for this example consisting of up to five MTS2BTSMI or MTS2BTA Bluetooth units connected through serial ports to workstations. One unit is configured as master and four units are configured as slaves. The master needs to be firmware version 3.6.2.1.0.0 or newer to support multipoint connections. The slaves are not firmware dependent and can be older models. All Bluetooth units are configured with baud rate of 9600bps, 8 data bits, no parity, 1 stop bit and hardware flow control. Baud rate can be set up to 115.2kbps but all Bluetooth devices need to be set to same baud rate.
Each workstation uses a terminal emulator like HyperTerminal or ZOC to communicate to Bluetooth device using AT commands.

Configure the slave Bluetooth devices to function as slaves in data mode.

- Verify local device is Slave in Data Mode.
  Sent: ATSI,7 <cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>0,1,0,0 <cr_lf>

- If not Slave, set to Slave in Data Mode and reset unit.
  Sent: ATSW25,0,1,0,0 <cr>
  Reply: OK<cr_lf>
  Sent: ATURST <cr>
  Reply: OK<cr_lf>

- Retrieve Bluetooth address from each Bluetooth slave device in this test setup.
  Sent: ATSI,1<cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>12-digit address <cr_lf>OK<cr_lf>

Issue these AT commands at the master Bluetooth device to generate four wireless connections between the single Master and the four slave devices.

- Verify local device is Master in Data Mode.
  Sent: ATSI,7 <cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>1,1,0,0 <cr_lf>

- If not Master, set to Master in Data Mode and reset unit.
  Sent: ATSW25,1,1,0,0 <cr>
  Reply: OK<cr_lf>
  Sent: ATURST <cr>
  Reply: OK<cr_lf>

- Enable multipoint mode and set number of connections to four.
  Sent: ATSSW,3,4 <cr>
  Reply: OK<cr_lf>

- Perform an Inquiry to detect ***BT_Addresses*** (unless it is already known).
  Sent: ATDI,4,00000000 {Class of Device} <cr> // Looks for four Bluetooth devices
  Reply: OK<cr_lf> // Search completed when DONE appears
  Reply: <cr_lf>BT_Slave1_Address,00000000,SOCKETWIRELESS<cr_lf>
  Reply: <cr_lf> BT_Slave2_Address,00000000,SOCKETWIRELESS<cr_lf>
  Reply: <cr_lf> BT_Slave3_Address,00000000,SOCKETWIRELESS<cr_lf>
  Reply: <cr_lf> BT_Slave4_Address,00000000,SOCKETWIRELESS<cr_lf>
  Reply: <cr_lf>DONE<cr_lf>

- Perform a Master Connection over SPP to the first Bluetooth slave device.
  Sent: ATDM, BT_Slave1_Address,1101<cr> // SPP connection
  Reply: OK<cr_lf>
  Reply: <cr_lf>CONNECT,00,BT_Slave1_Address <cr_lf> // Returns Slave BT address radio is in data mode

- Place radio back into Data Mode.
  Sent: +++ <cr> // Places Radio in Command Mode
  Reply: <cr_lf>OK<cr_lf>

- Perform a Master Connection over SPP to the second Bluetooth slave device.
  Sent: ATDM, BT_Slave2_Address,1101<cr> // SPP connection
  Reply: OK<cr_lf>
  Reply: <cr_lf>CONNECT,01,BT_Slave2_Address <cr_lf> // Returns Slave BT address radio is in data mode

- Place radio back into Data Mode.
  Sent: +++ <cr> // Places Radio in Command Mode
  Reply: <cr_lf>OK<cr_lf>

- Perform a Master Connection over SPP to the third Bluetooth slave device.
  Sent: ATDM, BT_Slave3_Address,1101<cr> // SPP connection
  Reply: OK<cr_lf>
  Reply: <cr_lf>CONNECT,02,BT_Slave3_Address <cr_lf> // Returns Slave BT address radio is in data mode

- Place radio back into Data Mode.
  Sent : +++ <cr> // Places Radio in Command Mode
  Reply: <cr_lf>OK<cr_lf>

- Perform a Master Connection over SPP to the fourth Bluetooth slave device.
  Sent: ATDM, BT_Slave4_Address,1101<cr> // SPP connection
  Reply: OK<cr_lf>
  Reply: <cr_lf>CONNECT,03,BT_Slave4_Address <cr_lf> // Returns Slave BT address radio is in data mode

Send and receive data between master and slave devices. Sending data from the master will be sent to all four slaves. Sending data from a slave will be sent only to the master.

Multi-Point Example Using MTS2BTSMI or MTS2BTA

Four Master and One Slave Devices

Note: Attaching the fourth master device can overload the connections and cause all connections to be terminated. Increasing the distance between Bluetooth devices will effect performance and could further limit the number of connections.

Setup hardware for this example consisting of up to five MTS2BTSMI or MTS2BTA Bluetooth units connected through serial ports to workstations. One unit is configured as slave and four units are configured as masters. The slave needs to be firmware version 3.6.2.1.0.0 or newer to support multipoint connections. The masters are not firmware dependent and can be older models. All Bluetooth units are configured with baud rate of 9600bps, 8 data bits, no parity, 1 stop bit and hardware flow control. Baud rate can be set up to 115.2kbps but all Bluetooth devices need to be set to same baud rate.

Each workstation uses a terminal emulator like HyperTerminal or ZOC to communicate to Bluetooth device using AT commands.

Configure the slave Bluetooth device to function as slave in data mode.

- Verify local device is Slave in Data Mode.
  Sent: ATSI,7 <cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>0,1,0,0 <cr_lf>

- If not Slave, set to Slave in Data Mode and reset unit.
  Sent: ATSW25,0,1,0,0 <cr>
  Reply: OK<cr_lf>

- Enable multipoint mode and set number of connections to four. Then reset unit.
  Sent: ATSSW,3,4 <cr>
  Reply: OK<cr_lf>
  Sent: ATURST <cr>
  Reply: OK<cr_lf>

- Retrieve Bluetooth address from Bluetooth slave device in this test setup.
  Sent: ATSI,1<cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>12-digit address <cr_lf>OK<cr_lf>

Issue these AT commands at the master Bluetooth devices to generate four wireless connections between the four Masters and the slave device.

- Verify local device is Master in Data Mode.
  Sent: ATSI,7 <cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>1,1,0,0 <cr_lf>

- If not Master, set to Master in Data Mode and reset unit.
  Sent: ATSW25,1,1,0,0 <cr>
  Reply: OK<cr_lf>
  Sent: ATURST <cr>
  Reply: OK<cr_lf>

- Perform an Inquiry to detect ***BT_Address*** of slave (unless it is already known).
  Sent: ATDI,1,00000000 {Class of Device} <cr> // Looks for one Bluetooth device
  Reply: OK<cr_lf> // Search completed when DONE appears
  Reply: <cr_lf>BT_Slave1_Address,00000000,SOCKETWIRELESS<cr_lf>
  Reply: <cr_lf>DONE<cr_lf>

- Perform a Master Connection over SPP to the Bluetooth slave device.
  Sent: ATDM, BT_Slave1_Address,1101<cr> // SPP connection
  Reply: OK<cr_lf>
  Reply: <cr_lf>CONNECT,BT_Slave1_Address <cr_lf> // Returns Slave BT address radio is in data mode

- At second master device perform a Master Connection over SPP to the Bluetooth slave device.
  Sent: ATDM, BT_Slave1_Address,1101<cr> // SPP connection
  Reply: OK<cr_lf>
  Reply: <cr_lf>CONNECT,BT_Slave1_Address <cr_lf> // Returns Slave BT address radio is in data mode

- At third master device perform a Master Connection over SPP to the Bluetooth slave device.
  Sent: ATDM, BT_Slave1_Address,1101<cr> // SPP connection
  Reply: OK<cr_lf>
  Reply: <cr_lf>CONNECT,BT_Slave1_Address <cr_lf> // Returns Slave BT address radio is in data mode

- At fourth master device perform a Master Connection over SPP to the Bluetooth slave device.
  Sent: ATDM, BT_Slave1_Address,1101<cr> // SPP connection
  Reply: OK<cr_lf>
  Reply: <cr_lf>CONNECT,BT_Slave1_Address <cr_lf> // Returns Slave BT address radio is in data mode

Send and receive data between master and slave devices. Sending data from the slave will be sent to all four masters. Sending data from a master device will be sent only to the slave device.

Repeater Example using MTS2BTSMI or MTS2BTA

Setup hardware for this example consisting of three MTS2BTSMI or MTS2BTA Bluetooth units connected through serial ports to workstations. One unit is configured as master, one unit is configured as slave and one unit is configured as repeater. The repeater needs to be firmware version 3.6.2.1.0.0 or newer to support repeater mode. The slave and master are not firmware dependent and can be older Multi-Tech models or generic Bluetooth devices. All Bluetooth units are configured with baud rate of 9600bps, 8 data bits, no parity, 1 stop bit and hardware flow control. Baud rate can be set up to 115.2kbps but all Bluetooth devices need to be set to same baud rate.

Each workstation uses a terminal emulator like HyperTerminal or ZOC to communicate to Bluetooth device using AT commands.

Configure the slave Bluetooth device to function as slave in data mode.

- Verify local device is Slave in Data Mode.
  Sent: ATSI,7 <cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>0,1,0,0 <cr_lf>

- If not Slave, set to Slave in Data Mode and reset unit.
  Sent: ATSW25,0,1,0,0 <cr>
  Reply: OK<cr_lf>
  Sent: ATURST <cr>
  Reply: OK<cr_lf>

- Retrieve Bluetooth address from the Bluetooth slave device in this test setup.
  Sent: ATSI,1<cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>12-digit slave address <cr_lf>OK<cr_lf>

Configure the repeater Bluetooth device to function as repeater in data mode.

- Enter address of slave device that repeater will automatically connect to once master connects to slave along with profile code.
  Sent: ATSMA,Slave_Address, 1101<cr>
  Reply: OK<cr_lf>

- Set repeater Bluetooth unit to function as repeater in Data Mode and reset unit.
  Sent: ATSW25,5,1,0,0 <cr>
  Reply: OK<cr_lf>
  Sent: ATURST <cr>
  Reply: OK<cr_lf>

- Retrieve Bluetooth address from the Bluetooth repeater device in this test setup.
  Sent: ATSI,1<cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>12-digit repeater address <cr_lf>OK<cr_lf>

Issue these AT commands at the master device to connect with the repeater which in turn connects with the slave device.

- Verify local device is Master in Data Mode.
  Sent: ATSI,7 <cr>
  Reply: OK<cr_lf>
  Reply: <cr_lf>1,1,0,0 <cr_lf>

- If not Master, set to Master in Data Mode and reset unit.
  Sent: ATSW25,1,1,0,0 <cr>
  Reply: OK<cr_lf>
  Sent: ATURST <cr>
  Reply: OK<cr_lf>

- Perform an Inquiry to detect ***BT_Addresses*** (unless it is already known).
  Sent: ATDI,2,00000000 {Class of Device} <cr> // Looks for two Bluetooth devices
  Reply: OK<cr_lf> // Search completed when DONE appears
  Reply: <cr_lf>BT_Slave1_Address,00000000,SOCKETWIRELESS<cr_lf>
  Reply: <cr_lf> BT_Repeater_Address,00000000,SOCKETWIRELESS<cr_lf>
  Reply: <cr_lf>DONE<cr_lf>

- Perform a Master Connection over SPP to the Bluetooth repeater device***.***
  Sent: ATDM, BT_Repeater_Address,1101<cr> // SPP connection
  Reply: OK<cr_lf>
  Reply: <cr_lf>CONNECT,00,BT_Repeater_Address <cr_lf> // Returns Repeater BT address radio is in data mode

Once Master connects to repeater the repeater will reply with CONNECT,00,BT_Master_Address and then repeater will automatically connect to slave and reply with CONNECT,01,BT_Slave_Address. Several NO ANSWER responses may display before connection to slave is completed.

Now the master and slave Bluetooth devices can send and receive data between each other through the repeater Bluetooth device. Data entered at the repeater workstation will be sent to both the master and slave units.

Appendix A– AT Command Summary Table

AT Command	Description	Requires Reset	Stores Permanently
Attention Prefix			
AT	Attention Prefix	N/A	N/A
Firmware Version			
ATVER,ver1	Module Firmware Version	N/A	Yes
Resetting			
ATURST	Unit Reset	N/A	N/A
ATFRST	Factory Reset	N/A	N/A
ATSSW,0	Set Bypass PIO(4) Factory Reconfiguration	Yes	Yes
ATRSW,0	Read Bypass PIO(4) Factory Reconfiguration	N/A	Yes
Boot Mode			
ATSSW,1	Set Boot Mode	Yes	Yes
ATRSW,1	Get Boot Mode	N/A	N/A
Security Level			
ATSSW,2	Set Security Level	Yes	Yes
ATRSW,2	Get Security Level	N/A	N/A
Get Status			
ATSI,0	Get Module Type	N/A	Yes
ATSI,1	Get Bluetooth Address	N/A	Yes
ATSI,2	Get Friendly Name	N/A	Yes
ATSI,3	Get Current Status of Connections	N/A	Yes
ATSI,4	Get Service Name	N/A	Yes
ATSI,5	Get Class of Device (COD)	N/A	Yes
ATSI,6	Get Response, Security, Auto SCO, Filter Settings	N/A	Yes
ATSI,7	Get Connection, Comm, UART, Service Modes	N/A	Yes
ATSI,8	Get UART Settings	N/A	Yes
ATSI,9	Get Master Auto-Connect Address	N/A	Yes
ATSI,10	Get Slave Scan Intervals and Windows	N/A	Yes
ATSI,11	Get PIO(5) Pulse Rate	N/A	Yes
ATSI,12	Get Escape Character	N/A	Yes
ATSI,13	Get Timeout Settings	N/A	Yes
ATSI,14	Get Maximum TX Power Level	N/A	Yes
ATSI,15	Get PIN Lock Mode	N/A	Yes
ATSI,16	Get Deep Sleep Mode	N/A	Yes
ATSI,17	Get Sniff Settings	N/A	Yes
ATSI,18	Get Link Supervisory Timeout	N/A	Yes
ATSI,19	Get List of Paired or Secured Addresses	N/A	Yes
ATSI,20	Get Channel UUIDs	N/A	Yes
ATSI,21	Get Specific Transmission Channel	N/A	Yes
ATSI,22	Get FTP/OPP Connection Status	N/A	Yes
# of Connections			
ATSSW,3	Set Max Connection Number	Yes	Yes
ATRSW,3	Read Max Connection Number	N/A	Yes

AT Command	Description	Requires Reset	Stores Permanently
Radio Name			
ATSN	Set Radio Name	No	Yes
ATRRN	Read Remote Radio Name By BT Address	N/A	Yes
Service Name			
ATSSN	Set Service Name	Yes	Yes
ATSSNC	Set Service Name by Channel	Yes	Yes
ATRSN	Read Service Name	N/A	Yes
ATRSNC	Read Service Name by Channel	N/A	Yes
ATRRSN	Read Remote Service Name	N/A	Yes
Security PIN			
SSW,6	Set PIN Request Handling Mode	Yes	Yes
RSW,6	Read PIN Request Handling Mode	N/A	Yes
ATSP	Set PIN	No	Yes
ATOP	Overwrite PIN	No	Yes
ATPR	Respond to Manual PIN Request	N/A	No
COD			
ATSC	Set Class of Device (COD)	Yes	Yes
Write Memory			
ATSW,20	Switch 20: Write UART Settings	No	Optional
ATSW,21	Switch 21: Write Slave Scan Intervals & Windows	Yes	Yes
ATSW,22	Switch 22: Write PIO Direction	No	Optional
ATSW,23	Switch 23: Write PIO Level	No	Optional
ATSW,24	Switch 24: Write Default Settings	For Security	Yes
ATSW,25	Switch 25: Write Power Up Default Modes	Yes	Yes
ATSW,26	Switch 26: Lock User Settings	No	Yes
ATSW,27	Switch 27: Write LED Rate	No	Yes
ATSW,28	Switch 28: Write Inquiry Timeout Settings	No	Yes
ATSW,29	Switch 29: Write PIN Lock Mode	No	Yes
ATSW,30	Switch 30: Write Deep Sleep Mode	No	Yes
Read Memory			
ATSR21	Read PIO Level	N/A	N/A
Inquiry			
ATDI	Dial Inquiry	N/A	N/A
ATIL	Last Inquiry	N/A	Yes
Master Connect			
ATDM	Dial As Master	N/A	N/A
ATDC	Dial Channel	N/A	N/A
ATDL	Dial Last	N/A	Yes
ATLAST	Read Last Connected Address	N/A	Yes
Master Default			
ATSMA	Set Master Default Address	Yes	Yes
ATMACLR	Master Address Clear	No	Yes
Connect Slave			
ATDS	Dial As Slave	N/A	N/A
Disconnect			
ATDH	Dial Hang Up	N/A	N/A

AT Command	Description	Requires Reset	Stores Permanently
ATDHC	Dial Hang Up By Channel	N/A	N/A
Modes			
+++	Default Escape Character	N/A	N/A
ATSESC	Set Command Mode Escape Character	No	Yes
ATMD	Put Radio Into Data Mode	No	No
ATMF	Put Radio Into Fast Data Mode	No	No
Cancel			
ATUCL	Cancel (Idle Mode)	No	No
Pairing			
ATPAIR	Pair Radios	No	Yes
ATUPAIR	Unpair By Index	No	Yes
ATUPAIRB	Unpair By Bluetooth Address	No	Yes
ATCPAIR	Clear all paired or secured connections	No	Yes
Sniff and Park			
ATSNIFF	Enable Sniff	No	Yes
ATSSNIFF	Enable Auto Sniff	No	Yes
ATCSNIFF	Clear Sniff	No	Yes
ATXSNIFF	Exit Sniff	No	N/A
ATPARK	Park	No	No
ATXPARK	Exit Park	No	N/A
RSSI and Link			
ATRSSI	Get RSSI Value	N/A	No
ATRSSIC	Get RSSI Value by Channel	N/A	No
ATLQ	Get Link Quality	N/A	No
ATLQC	Get Link Quality by Channel	N/A	No
Audio PCM			
ATDSCO	Dial SCO	N/A	N/A
ATDHSCO	Dial Hang Up SCO	N/A	N/A
Max TX Power			
ATSPF	Set Max TX Power Level	No	Yes
Link Timeout			
ATLSTO	Link Supervisory Timeout	No	Yes
Variable Storage			
ATSTORE	Store Variable	No	Yes
ATREAD	Read Variable	N/A	Yes
Conn PIO Logic			
ATSSW,4	Set Connection PIO Logic	No	Yes
Transmission Ch			
ATSWC	Select Transmission Channel	No	No
Discoverability			
ATSDIS	Set Discoverable	No	No
ATUDIS	Set Undiscoverable	No	No
FTP Client			

AT Command	Description	Requires Reset	Stores Permanently
ATDFTP	Dial FTP	No	No
ATDHFT	Dial Hang Up FTP	No	No
ATFTPB	FTP Browse	No	No
ATFTPSUB	Move to FTP Sub-Directory	No	No
ATFTPUP	Move to FTP Parent Directory	No	No
ATFTPROOT	Move to Root Directory	No	No
ATFTPPUSH	FTP Push	No	No
ATFTPPULL	FTP Pull	No	No
OPP Client			
ATDOPP	Dial OPP	No	No
ATDHOPP	Dial Hang Up OPP	No	No
ATWBCARD	Push Business Card	No	No
ATRBCARD	Pull Business Card	No	No
ATOPPUSH	Move	No	No
OPP Server			
ATBCARDR	Handle Business Card Request	No	No

Appendix B – Responses

Notes on Short Response Mode Structure:

- The first number (2-3 digits) returned identifies the command that was sent. All identifiers are unique to each specific command except for commands such as ATSSN and ATSSNC. These have the same basic function, only ATSSN is for point-to-point and ATSSNC is for multi-point use. In this case, the identifiers will be the same.
- The second number (2 digits) returned after the comma is the response code. If the response code is 00 ("OK" in long response mode), then the command has been successfully received by the radio. This does not mean the command is complete, but just that it has been accepted by the radio, as some commands may not go into effect or return data immediately. Anything other than a 00 returned in the second position means there was an error with the command.
- Any data returned after the second number will vary based on the command entered. See the response table below for data formatting.

Response Table:

All example responses shown are with factory default settings in place. See AT Commands section for description of returned parameters.

AT Command	Example Long Response	Example Short Response
Attention Prefix		
AT	<cr_lf>OK<cr_lf>	<cr_lf>107,00<cr_lf>
Firmware Version		
ATVER,ver1	<cr_lf>OK<cr_lf><cr_lf>Ver 3.6.2.1.0.0<cr_lf>	<cr_lf>52,00,Ver 3.6.2.1.0.0<cr_lf>
Resetting		
ATURST	No Response	No Response
ATFRST	<cr_lf>OK<cr_lf><cr_lf>RESET COMPLETE<cr_lf>	<cr_lf>75,00<cr_lf><cr_lf>RESET COMPLETE<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>75,01<cr_lf>
ATSSW,0	<cr_lf>OK<cr_lf>	<cr_lf>102,00 <cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>102,01<cr_lf>
ATRSW,0	<cr_lf>OK<cr_lf><cr_lf>0<cr_lf>	<cr_lf>103,00,00<cr_lf>
Boot Mode		
ATSSW,1	<cr_lf>OK<cr_lf>	<cr_lf>102,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>102,01<cr_lf>
ATRSW,1	<cr_lf>OK<cr_lf>	<cr_lf>103,00,00<cr_lf>
Security Level		
ATSSW,2	<cr_lf>OK<cr_lf>	<cr_lf>102,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>102,01<cr_lf>
ATRSW,2	<cr_lf>OK<cr_lf>	<cr_lf>103,00,00<cr_lf>
Get Status		
ATSI,1	<cr_lf>OK<cr_lf><cr_lf>00A09608F513<cr_lf>	<cr_lf>14,00,123456789012<cr_lf>
ATSI,2	<cr_lf>OK<cr_lf><cr_lf>SocketWireless<cr_lf>	<cr_lf>53,00,0A,SocketWireless<cr_lf>
ATSI,3	<cr_lf>OK<cr_lf><cr_lf>0,0<cr_lf>	<cr_lf>17,00,0,0<cr_lf>

AT Command	Example Long Response	Example Short Response
ATSI,4	<cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>	<cr_lf>18,00,COM0<cr_lf>
ATSI,5	<cr_lf>OK<cr_lf><cr_lf>00000000<cr_lf>	<cr_lf>19,00,00000000<cr_lf>
ATSI,6	<cr_lf>OK<cr_lf><cr_lf>0,0,0,0<cr_lf>	<cr_lf>20,00,1,0,0,0<cr_lf>
ATSI,7	<cr_lf>OK<cr_lf><cr_lf>0,1,0,0<cr_lf>	<cr_lf>21,00,0,1,0,0<cr_lf>
ATSI,8	<cr_lf>OK<cr_lf><cr_lf>0027,0000,0000<cr_lf>	<cr_lf>22,00,0027,0000,0000<cr_lf>
ATSI,9	<cr_lf>OK<cr_lf><cr_lf>Not Set!<cr_lf>	<cr_lf>23,00,Not Set!<cr_lf>
ATSI,10	<cr_lf>OK<cr_lf><cr_lf>0400,0200,0400,0200<cr_lf>	<cr_lf>24,00,0400,0200,0400,0200<cr_lf>
ATSI,11	<cr_lf>OK<cr_lf><cr_lf>03E8<cr_lf>	<cr_lf>25,00,03E8<cr_lf>
ATSI,12	<cr_lf>OK<cr_lf><cr_lf>2B<cr_lf>	<cr_lf>68,00,2B<cr_lf>
ATSI,13	<cr_lf>OK<cr_lf><cr_lf>0010,0028,003C<cr_lf>	<cr_lf>69,00,0010,0028,003C<cr_lf>
ATSI,14	<cr_lf>OK<cr_lf><cr_lf>default<cr_lf>	<cr_lf>71,01<cr_lf>
ATSI,15	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>	<cr_lf>73,00,00<cr_lf>
ATSI,16	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>	<cr_lf>76,00,00<cr_lf>
ATSI,17	<cr_lf>OK<cr_lf><cr_lf>Not Set!<cr_lf>	<cr_lf>92,00,Not Set!<cr_lf>
ATSI,18	<cr_lf>OK<cr_lf><cr_lf>04<cr_lf>	<cr_lf>93,00,04<cr_lf>
ATSI,19	<cr_lf>OK<cr_lf><cr_lf>00,<cr_lf>01,<cr_lf>02,<cr_lf>03,<cr_lf>	<cr_lf>94,00<cr_lf>00,<cr_lf>01,<cr_lf>02,<cr_lf>03,<cr_lf>
ATSI,20	<cr_lf>OK<cr_lf><cr_lf>1101<cr_lf>	<cr_lf>110,00,1101lf>
ATSI,21	<cr_lf>OK<cr_lf><cr_lf>00,01<cr_lf>	<cr_lf>127,00,00,00<cr_lf>
ATSI,22	<cr_lf>OK<cr_lf><cr_lf>1<cr_lf>	<cr_lf>131,00,1<cr_lf>
# of Connections		
ATSSW,3	<cr_lf>OK<cr_lf>	<cr_lf>102,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>102,01<cr_lf>
ATRSW,3	<cr_lf>OK<cr_lf><cr_lf>01<cr_lf>	<cr_lf>103,00,01<cr_lf>
Radio Name		
ATSN	<cr_lf>OK<cr_lf>	<cr_lf>15,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>15,02<cr_lf>
ATRRN	<cr_lf>OK<cr_lf><cr_lf>SocketWireless<cr_lf>	<cr_lf>109,00,0A,SocketWireless<cr_lf>
	<cr_lf>OK<cr_lf><cr_lf>NO ANSWER<cr_lf>	<cr_lf>109,01<cr_lf>
Service Name		
ATSSN	<cr_lf>OK<cr_lf>	<cr_lf>38,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>38,03<cr_lf>
ATSSNC	<cr_lf>OK<cr_lf>	<cr_lf>38,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>38,03<cr_lf>
ATRSN	<cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>	<cr_lf>18,00,04,COM0<cr_lf>
ATRSNC	<cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>	<cr_lf>18,00,04,COM0<cr_lf>
ATRRSN	<cr_lf>OK<cr_lf><cr_lf>01,Bluetooth Serial Port<cr_lf>	<cr_lf>108,00,16,Bluetooth Serial Port<cr_lf>
	<cr_lf>OK<cr_lf><cr_lf>NO ANSWER<cr_lf>	<cr_lf>108,01<cr_lf>
Security PIN		
SSW,6	<cr_lf>OK<cr_lf>	<cr_lf>102,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>102,01<cr_lf>
RSW,6	<cr_lf>OK<cr_lf><cr_lf>01<cr_lf>	<cr_lf>103,00,01<cr_lf>
ATSP	<cr_lf>OK<cr_lf>	<cr_lf>39,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>39,02<cr_lf>
ATOP	<cr_lf>OK<cr_lf>	<cr_lf>78,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>78,03<cr_lf>
ATPR	<cr_lf>OK<cr_lf>	<cr_lf>132,00<cr_lf>
COD		
ATSC	<cr_lf>OK<cr_lf>	<cr_lf>40,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>40,02<cr_lf>

AT Command	Example Long Response	Example Short Response
Write Memory		
ATSW20	No Response	No Response
ATSW21	<cr_lf>OK<cr_lf>	<cr_lf>47,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>47,01<cr_lf>
ATSW22	<cr_lf>OK<cr_lf>	<cr_lf>48,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>48,01<cr_lf>
ATSW23	<cr_lf>OK<cr_lf>	<cr_lf>49,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>49,01<cr_lf>
ATSW24	<cr_lf>OK<cr_lf>	<cr_lf>34,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>34,01<cr_lf>
ATSW25	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>35,01<cr_lf>
ATSW26	<cr_lf>OK<cr_lf>	<cr_lf>36,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>36,01<cr_lf>
ATSW27	<cr_lf>OK<cr_lf>	<cr_lf>46,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>46,01<cr_lf>
ATSW28	<cr_lf>OK<cr_lf>	<cr_lf>67,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>67,01<cr_lf>
ATSW29	<cr_lf>OK<cr_lf>	<cr_lf>72,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>72,01<cr_lf>
ATSW30	<cr_lf>OK<cr_lf>	<cr_lf>74,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>74,01<cr_lf>
Read Memory		
ATSR21	<cr_lf>OK<cr_lf><cr_lf>0<cr_lf>	<cr_lf>44,00,0<cr_lf>
Inquiry		
ATDI	<cr_lf>OK<cr_lf><cr_lf>123456789012,12345678<cr_lf><cr_lf>DONE<cr_lf>	<cr_lf>12,00<cr_lf><cr_lf>13,123456789012,12345678<cr_lf><cr_lf>51,01<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>12,01<cr_lf>
ATIL	<cr_lf>OK<cr_lf><cr_lf>123456789012,12345678<cr_lf><cr_lf>DONE<cr_lf>	<cr_lf>87,00<cr_lf><cr_lf>13,123456789012,12345678<cr_lf><cr_lf>51,01<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>87,01<cr_lf>
Master Connect		
ATDM	<cr_lf>OK<cr_lf><cr_lf>CONNECT,123456789012<cr_lf>	<cr_lf>21,00,123456789012<cr_lf>
(w/Security)	<cr_lf>OK<cr_lf><cr_lf>PAIRED,123456789012<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	<cr_lf>82,123456789012<cr_lf><cr_lf>21,00,123456789012<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>21,02<cr_lf>
	<cr_lf>NO ANSWER<cr_lf>	<cr_lf>21,04<cr_lf>
ATDC	<cr_lf>OK<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	<cr_lf>21,00,123456789012<cr_lf>
(w/Security)	<cr_lf>OK<cr_lf><cr_lf>PAIRED,123456789012<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	<cr_lf>82,123456789012<cr_lf><cr_lf>21,00,123456789012<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>21,02<cr_lf>
	<cr_lf>NO ANSWER<cr_lf>	<cr_lf>21,04<cr_lf>
ATDL	<cr_lf>OK<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	<cr_lf>77,00<cr_lf><cr_lf>21,00,123456789012<cr_lf>
(w/Security)	<cr_lf>OK<cr_lf><cr_lf>PAIRED,123456789012<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	<cr_lf>77,00<cr_lf><cr_lf>82,123456789012<cr_lf><cr_lf>21,00,123456789012<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>77,02<cr_lf>
	<cr_lf>NO ANSWER<cr_lf>	<cr_lf>77,04<cr_lf>
ATLAST	<cr_lf>OK<cr_lf><cr_lf>123456879012<cr_lf>	cr_lf>64,00,123456789012<cr_lf>

AT Command	Example Long Response	Example Short Response
Master Default		
ATSMA	<cr_lf>OK<cr_lf>	<cr_lf>42,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>42,01<cr_lf>
ATMACLR	<cr_lf>OK<cr_lf>	<cr_lf>43,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>43,01<cr_lf>
ATSSW,7,1	<cr_lf>OK<cr_lf>	<cr_lf>102,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>102,01<cr_lf>
RSW,7	<cr_lf>OK<cr_lf><cr_lf>01<cr_lf>	<cr_lf>103,00,01<cr_lf>
Connect Slave		
ATDS	<cr_lf>OK<cr_lf>	<cr_lf>22,00<cr_lf>
Disconnect		
ATDH	<cr_lf>OK<cr_lf><cr_lf>DISCONNECT<cr_lf>	<cr_lf>23,00<cr_lf>
	<cr_lf>OK<cr_lf><cr_lf>LINKLOSS<cr_lf>	<cr_lf>23,01<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>23,02<cr_lf>
ATDHC	<cr_lf>OK<cr_lf><cr_lf>DISCONNECT,00<cr_lf>	<cr_lf>23,00,00<cr_lf>
	<cr_lf>OK<cr_lf><cr_lf>LINKLOSS,00<cr_lf>	<cr_lf>23,01,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>23,03<cr_lf>
Modes		
+++	<cr_lf>OK<cr_lf>	<cr_lf>32,00<cr_lf>
ATSESC	<cr_lf>OK<cr_lf>	<cr_lf>65,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>65,01<cr_lf>
ATMD	<cr_lf>OK<cr_lf>	<cr_lf>31,00<cr_lf>
	<cr_lf>NO CARRIER<cr_lf>	<cr_lf>31,03<cr_lf>
ATMF	<cr_lf>OK<cr_lf>	<cr_lf>33,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>33,01<cr_lf>
	<cr_lf>NO CARRIER<cr_lf>	<cr_lf>33,02<cr_lf>
Cancel		
ATUCL	<cr_lf>OK<cr_lf>	<cr_lf>51,00<cr_lf>
Pairing		
ATPAIR	<cr_lf>OK<cr_lf><cr_lf>PAIRED,123456789012<cr_lf>	<cr_lf>70,00<cr_lf><cr_lf>82,123456789012<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>70,01<cr_lf>
	<cr_lf>PAIRED,FAILED<cr_lf>	<cr_lf>70,02<cr_lf>
ATUPAIR	<cr_lf>OK<cr_lf>	<cr_lf>80,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>80,02<cr_lf>
ATUPAIRB	<cr_lf>OK<cr_lf>	<cr_lf>96,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>96,02<cr_lf>
ATCPAIR	<cr_lf>OK<cr_lf>	<cr_lf>97,00<cr_lf>
Sniff and Park		
ATSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>27,00<cr_lf>
	<cr_lf>NO CARRIER<cr_lf>	<cr_lf>27,02<cr_lf>
ATSSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>94,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>94,01<cr_lf>
ATCSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>95,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>95,01<cr_lf>
ATXSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>28,00<cr_lf>
	<cr_lf>NO CARRIER<cr_lf>	<cr_lf>28,02<cr_lf>
ATPARK	<cr_lf>OK<cr_lf>	<cr_lf>26,00<cr_lf>
	<cr_lf>NO CARRIER<cr_lf>	<cr_lf>26,02<cr_lf>
ATXPARK	<cr_lf>OK<cr_lf>	<cr_lf>29,00<cr_lf>
	<cr_lf>NO CARRIER<cr_lf>	<cr_lf>29,02<cr_lf>

AT Command	Example Long Response	Example Short Response
RSSI and Link		
ATRSSI	<cr_lf>OK<cr_lf><cr_lf>-10<cr_lf>	<cr_lf>100,00,+00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>100,02<cr_lf>
ATRSSIC	<cr_lf>OK<cr_lf><cr_lf>-10<cr_lf>	<cr_lf>100,00,+00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>100,02<cr_lf>
ATLQ	<cr_lf>OK<cr_lf><cr_lf>FF<cr_lf>	<cr_lf>101,00,FF<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>101,02<cr_lf>
ATLQC	<cr_lf>OK<cr_lf><cr_lf>FF<cr_lf>	<cr_lf>101,00,FF<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>101,02<cr_lf>
Max TX Power		
ATSPF	<cr_lf>OK<cr_lf>	<cr_lf>79,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>79,01<cr_lf>
Link Timeout		
ATLSTO	<cr_lf>OK<cr_lf>	<cr_lf>88,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>88,01<cr_lf>
Variable Storage		
ATSTORE	<cr_lf>OK<cr_lf>	<cr_lf>90,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>90,01<cr_lf>
ATREAD	<cr_lf>OK<cr_lf>	<cr_lf>91,00<cr_lf>
Variable Storage		
ATSTORE	<cr_lf>OK<cr_lf>	<cr_lf>90,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>90,01<cr_lf>
ATREAD	<cr_lf>OK<cr_lf>	<cr_lf>91,00<cr_lf>
Variable Storage		
ATSTORE	<cr_lf>OK<cr_lf>	<cr_lf>90,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>90,01<cr_lf>
ATREAD	<cr_lf>OK<cr_lf>	<cr_lf>91,00<cr_lf>
Conn PIO Logic		
ATSSW,4	<cr_lf>OK<cr_lf>	<cr_lf>102,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>102,01<cr_lf>
ATRSW,4	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>	<cr_lf>103,00,00<cr_lf>
Transmission Ch		
ATSWC	<cr_lf>OK<cr_lf>	<cr_lf>121,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>121,03<cr_lf>
Discoverability		
ATSDIS	<cr_lf>OK<cr_lf>	<cr_lf>114,00<cr_lf>
ATUDIS	<cr_lf>OK<cr_lf>	<cr_lf>115,00<cr_lf>
FTP Con/Discon		
ATDFTP	<cr_lf>OK<cr_lf><cr_lf>FTP CONNECT,123456789012<cr_lf>	<cr_lf>111,00,123456789012<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>111,01<cr_lf>
	<cr_lf>FTP NO ANSWER<cr_lf>	<cr_lf>111,03<cr_lf>
ATDHFTP	<cr_lf>OK<cr_lf><cr_lf>FTP DISCONNECT<cr_lf>	<cr_lf>112,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>112,01<cr_lf>

AT Command	Example Long Response	Example Short Response
FTP Directory		
ATFTPB	<cr_lf>OK<cr_lf><XML Directory Information><cr_lf>BROWSE COMPLETE<cr_lf>	<cr_lf>112,00<cr_lf><XML Directory Information><cr_lf>113,01<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>113,02<cr_lf>
ATFTPSUB	<cr_lf>OK<cr_lf>	<cr_lf>116,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>116,01<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>116,02<cr_lf>
ATFTPUP	<cr_lf>OK<cr_lf>	<cr_lf>118,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>118,01<cr_lf>
ATFTPROOT	<cr_lf>OK<cr_lf>	<cr_lf>117,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>117,01<cr_lf>
FTP Push/Pull		
ATFTPPUSH	<cr_lf>OK<cr_lf><cr_lf>SENT,0001<cr_lf><cr_lf>SENT,0051<cr_lf><cr_lf>PUSH COMPLETE<cr_lf>	<cr_lf>120,00<cr_lf><cr_lf>120,01,0001<cr_lf><cr_lf>120,01,0051<cr_lf><cr_lf>120,02<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>120,03<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>120,04<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>120,05<cr_lf>
ATFTPPULL	<cr_lf>OK<cr_lf><cr_lf>File,0030,<30 Bytes of Data><cr_lf><cr_lf>PULL COMPLETE<cr_lf>	<cr_lf>119,00,001E,<30 Bytes of Data><cr_lf><cr_lf>119,01<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>119,03<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>119,04<cr_lf>
FTP Server		
File Push	<cr_lf>FTP CONNECT,0123456789012<cr_lf><cr_lf>PUSH START,<File Name>,<Size><cr_lf><Data><cr_lf><cr_lf>PUSH COMPLETE <cr_lf><cr_lf>FTP<cr_lf>	<cr_lf>111,00,123456789012<cr_lf><cr_lf>129,01,<File Name>,<Size><cr_lf><Data><cr_lf><cr_lf>129,02<cr_lf><cr_lf>112,00<cr_lf>
OPP Con/Discon		
ATDOPP	<cr_lf>OK<cr_lf><cr_lf>OPP CONNECT,123456789012<cr_lf>	<cr_lf>122,00,123456789012<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>122,01<cr_lf>
	<cr_lf> OPP NO ANSWER<cr_lf>	<cr_lf>122,03<cr_lf>
ATDHOPP	<cr_lf>OK<cr_lf><cr_lf> OPP DISCONNECT<cr_lf>	<cr_lf>123,00<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>123,01<cr_lf>
OPP Push/Pull		
ATSBCARD	<cr_lf>OK<cr_lf><vCard Data><<cr_lf>PUSH COMPLETE<cr_lf>	<cr_lf>125,00<cr_lf><vCard Data><cr_lf>125,02<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>125,03<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>125,04<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>125,05<cr_lf>
ATRBCARD	<cr_lf>OK<cr_lf><cr_lf><vCard Data><cr_lf><cr_lf>PULL COMPLETE<cr_lf>	<cr_lf>126,00,<vCard Data><cr_lf><cr_lf>126,01<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>126,02<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>126,03<cr_lf>
ATOPPPUSH	<cr_lf>OK<cr_lf><cr_lf>PUSH COMPLETE<cr_lf>	<cr_lf>124,00<cr_lf><cr_lf>124,02<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>124,03<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>124,04<cr_lf>

AT Command	Example Long Response	Example Short Response
OPP Server		
ATBCARDR	<cr_lf>OK<cr_lf><vCard Data><cr_lf>PULL COMPLETE<cr_lf>	<cr_lf>128,00<cr_lf><vCard Data><cr_lf>128,05<cr_lf>
	<cr_lf>ERROR<cr_lf>	<cr_lf>128,03<cr_lf>
Business Card Push	<cr_lf>OPP CONNECT,123456789012<cr_lf> <cr_lf>PUSH START,<File Name>,<Size> <cr_lf><cr_lf><Data><cr_lf><cr_lf>PUSHCOMPLETE<cr_lf><cr_lf>OPP DISCONNECT<cr_lf>	<cr_lf>122,00,123456789012<cr_lf> <cr_lf>124,01,<File Name>,<Size> <cr_lf>data<cr_lf>124,02<cr_lf><cr_lf>123,00<cr_lf>

Appendix C -- UUID Table

The Highlighted Universal Unique Identifiers (UUID's) have been tested as both Client and Server. Others may be activated, but the results are unknown.

Profile Name	UUID
Serial Port Profile (SPP)	1101
LAN Accessing PPP	1102
Dial-up Network (DUN)	1103
IrMC Sync	1104
OPP - Object Push/Pull	1105
OBEX File Transfer (Client) only	1106
IrMC Sync Command	1107
Headset	1108
Cordless Telephone (CTP)	1109
Intercom	1110
Fax	1111
Audio Gateway	1112
WAP	1113
WAP_CLIENT	1114
BNEP/PAN (Client)	0000

Appendix D – Class 1 Module Power Consumption

Idle Mode ≈ 1.4mA average

Deep Sleep Mode ≈ 25-50uA (Idle Mode)
- If in Sniff Mode, will go momentarily as low as 50uA in between processes.

Slave Unconnected ≈ 39mA average
- ATSW21,4096,18,4096,18 settings will result in the Slave not connected ≈ 1mA average.
- ATSW21,4096,18,4096,18 settings along with enabling Deep Sleep Mode will result in an unconnected slave current draw of ≈ 350uA average.

Master Unconnected ≈ 1.4mA average
- A Master device not in use – it is better to just turn off the power completely to the device and draw zero current.

Master Inquiry ≈ 60mA average
- For the first couple of seconds before it connects to the Slave

Connected ≈ 5mA average
- ATSNIFF,1600,160,10,160 will result in the Slave connected ≈ 1.4mA average, with no data being sent.
- This can even go as low as 0.6mA if you make the Slave not discoverable but connectable.

If you perform a remote *Bluetooth* RF "ATDH" disconnect command from any Master to the Slave, the Slave will go into idle mode. You will have to either send a command to the Slave through its local UART or cycle power on the device to have it come back up in Slave mode. This was designed as if you were communicating to the local UART on the Slave device. Typically, a Slave never sends a disconnect command; only the Master does. In this mode, you can devise some clever power-saving features such as leaving the Slave in idle mode until another event triggers the device to go back into Slave discoverable mode. t is important to remember that a device in Idle Mode is not discoverable; this has some security advantages. The same benefits can be achieved by simply controlling when the device is turned off/on again.

Appendix E – Two Versions of Bluetooth Command Responses Comparison

This Appendix compares the responses for Bluetooth Version 3.6.2.1.0.0 with Bluetooth Version 2.8.1.1.0.0. All example responses shown here are with factory default settings in place.

Command	Version 3.6.2.1.0.0	Version 2.8.1.1.0.0
Attention Prefix		
AT	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
Firmware Version		
ATVER,ver1	<cr_lf>OK<cr_lf><cr_lf>Ver 3.6.2.1.0.0<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>Ver 2.8.1.1.0<cr_lf>
Resetting		
ATURST	No Response Displayed	No Response Displayed
ATFRST	<cr_lf>OK<cr_lf><cr_lf>RESET COMPLETE<cr_lf>	No Response Displayed
ATSSW,0	<cr_lf>OK<cr_lf>	Not Applicable (N.A.)
ATRSW,0	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>	N.A.
Get Status		
ATSI,1	<cr_lf>OK<cr_lf><cr_lf>00A09608F513<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>00A09608F513<cr_lf>
ATSI,2	<cr_lf>OK<cr_lf><cr_lf>SocketWireless<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>SocketWireless<cr_lf>
ATSI,3	<cr_lf>OK<cr_lf><cr_lf>1,0,0,0,0<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>1,0<cr_lf>
ATSI,4	<cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>
ATSI,5	<cr_lf>OK<cr_lf><cr_lf>00000000<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>00000000<cr_lf>
ATSI,6	<cr_lf>OK<cr_lf><cr_lf>0,0,0,0<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>0,0,0,0<cr_lf>
ATSI,7	<cr_lf>OK<cr_lf><cr_lf>0,1,0,0<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>0,1,0,0<cr_lf>
ATSI,8	<cr_lf>OK<cr_lf><cr_lf>0027,0000,0000<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>0027<cr_lf>
ATSI,9	<cr_lf>OK<cr_lf><cr_lf>Not Set!<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>Not Set!<cr_lf>
ATSI,10	<cr_lf>OK<cr_lf><cr_lf>0400,0200,0400,0200<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>0400,0200,0400,0200<cr_lf>
ATSI,11	<cr_lf>OK<cr_lf><cr_lf>03E8<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>04B0<cr_lf>
ATSI,12	<cr_lf>OK<cr_lf><cr_lf>2B<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>2B<cr_lf>
ATSI,13	<cr_lf>OK<cr_lf><cr_lf>0010,0028,003C<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>003C,003C<cr_lf>
ATSI,14	<cr_lf>OK<cr_lf><cr_lf>default<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>default<cr_lf>
ATSI,15	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>
ATSI,16	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>
ATSI,17	<cr_lf>OK<cr_lf><cr_lf>Not Set!<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>Not Set!<cr_lf>
ATSI,18	<cr_lf>OK<cr_lf><cr_lf>04<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>04<cr_lf>
ATSI,19	<cr_lf>OK<cr_lf><cr_lf>00,<cr_lf>01,<cr_lf>02,<cr_lf>03,<cr_lf>	N.A.
ATSI,20	<cr_lf>OK<cr_lf><cr_lf>1101<cr_lf>	N.A.
ATSI,21	<cr_lf>OK<cr_lf><cr_lf>00,01<cr_lf>	N.A.
ATSI,22	<cr_lf>OK<cr_lf><cr_lf>1<cr_lf>	N.A.
# of Connections		
ATSSW	<cr_lf>OK<cr_lf>	N.A.
ATRSW	<cr_lf>OK<cr_lf><cr_lf>01<cr_lf>	N.A.
Radio Name		
ATSN	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATRRN	<cr_lf>OK<cr_lf><cr_lf>SocketWireless <cr_lf>	N.A.

Command	Version 3.6.2.1.0.0	Version 2.8.1.1.0.0
Service Name		
ATSSN	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSSNC	<cr_lf>OK<cr_lf>	N.A.
ATRSN	<cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>	N.A.
ATRSNC	<cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>	N.A.
ATRRSN	<cr_lf>OK<cr_lf><cr_lf>01,Bluetooth Serial Port<cr_lf>	N.A.
Security PIN		
SSW,6	<cr_lf>OK<cr_lf>	N.A.
RSW,6	<cr_lf>OK<cr_lf><cr_lf>01<cr_lf>	N.A.
ATSP	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATOP	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATPR	<cr_lf>OK<cr_lf>	N.A.
COD		
ATSC	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
Write Memory		
ATSW20,X,X,X,X	No Response Displayed	No Response Displayed
ATSW21,X,X,X,X	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSW22,X,X,X	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSW23,X,X,X	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSW24,X,X,X,X	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSW25,X,X,X,X	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSW26,X,X	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSW27,X	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSW28,X,X	N.A.	<cr_lf>OK<cr_lf>
ATSW28,X,X,X	<cr_lf>OK<cr_lf>	N.A.
ATSW29,X,X	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSW30,X	<cr_lf>OK<cr_lf>	N.A.
Read Memory		
ATSR21	<cr_lf>OK<cr_lf><cr_lf>0<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>0<cr_lf>
Inquiry		
ATDI	<cr_lf>OK<cr_lf><cr_lf>BT_Address, COD<cr_lf><cr_lf>DONE<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>BT_Address, COD,Name<cr_lf><cr_lf>DONE<cr_lf>
ATIL	<cr_lf>OK<cr_lf><cr_lf>BT_Address, COD<cr_lf><cr_lf>DONE<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>BT_Address, COD,Name<cr_lf><cr_lf>DONE<cr_lf>
Master Connect		
ATDM	<cr_lf>OK<cr_lf><cr_lf>CONNECT,123456789012<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>CONNECT,123456789012<cr_lf>
(w/Security)	<cr_lf>OK<cr_lf><cr_lf>PAIRED,123456789012<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	<cr_lf>LINK,123456789012<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>
ATDC	<cr_lf>OK<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	N.A.
(w/Security)	<cr_lf>OK<cr_lf><cr_lf>PAIRED,123456789012<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	N.A.
ATDL	<cr_lf>OK<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>
(w/Security)	<cr_lf>OK<cr_lf><cr_lf>PAIRED,123456789012<cr_lf><cr_lf>CONNECT ,123456789012<cr_lf>	N.A.
ATLAST	<cr_lf>OK<cr_lf><cr_lf>123456879012 <cr_lf>	<cr_lf>OK<cr_lf><cr_lf>123456879012<cr_lf>

Command	Version 3.6.2.1.0.0	Version 2.8.1.1.0.0
Master Default		
ATSMA	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATMACLR	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSSW,7,1	<cr_lf>OK<cr_lf>	N.A.
ATRSW,7	<cr_lf>OK<cr_lf><cr_lf>01<cr_lf>	N.A.
Connect Slave		
ATDS	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
Disconnect		
ATDH	<cr_lf>OK<cr_lf><cr_lf>DISCONNECT <cr_lf>	<cr_lf>OK<cr_lf><cr_lf>NO CARRIER<cr_lf>
ATDHC	<cr_lf>OK<cr_lf><cr_lf>DISCONNECT,00 <cr_lf>	N.A.
Modes		
+++	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSESC	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATMD	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATMF	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
Cancel		
ATUCL	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
Pairing		
ATPAIR	<cr_lf>OK<cr_lf><cr_lf>PAIRED,123456789 012<cr_lf>	<cr_lf>OK<cr_lf><cr_lf>PAIRED,12345678901 2<cr_lf>
ATAPAIR	N.A.	<cr_lf>OK<cr_lf><cr_lf>123456789012<cr_lf>
ATUPAIR	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATUPAIRB	<cr_lf>OK<cr_lf>	N.A.
ATCPAIR	<cr_lf>OK<cr_lf>	N.A.
Sniff and Park		
ATSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATSSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATCSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATXSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATPARK	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATXPARK	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
RSSI and Link		
ATRSSI	<cr_lf>OK<cr_lf><cr_lf>-10<cr_lf>	N.A.
ATRSSIC	<cr_lf>OK<cr_lf><cr_lf>-10<cr_lf>	N.A.
ATLQ	<cr_lf>OK<cr_lf><cr_lf>FF<cr_lf>	N.A.
ATLQC	<cr_lf>OK<cr_lf><cr_lf>FF<cr_lf>	N.A.
Audio PCM – only on voice build		
ATDSCO	<cr_lf>OK<cr_lf><cr_lf>SCO CONNECT<cr_lf>	N.A.
ATDHSCO	<cr_lf>OK<cr_lf><cr_lf>SCO DISCONNECT<cr_lf>	N.A.
Max TX Power		
ATSPF	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
Link Timeout		
ATLSTO	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
Variable Storage		
ATSTORE	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>
ATREAD	<cr_lf>OK<cr_lf>	<cr_lf>OK<cr_lf>

AT Command	Version 3.6.2.1.0.0	Version 2.8.1.1.0.0
Variable Storage		
ATSTORE	<cr_lf>OK<cr_lf>	N.A.
ATREAD	<cr_lf>OK<cr_lf>	N.A.
Conn PIO Logic		
ATSSW,4	**<cr_lf>OK<cr_lf>**	N.A.
ATRSW,4	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>	N.A.
Transmission Ch		
ATSWC	<cr_lf>OK<cr_lf>	N.A.
Discoverability		
ATSDIS	<cr_lf>OK<cr_lf>	N.A.
ATUDIS	<cr_lf>OK<cr_lf>	N.A.
FTP Con/Discon		
ATDFTP	<cr_lf>OK<cr_lf><cr_lf>FTP CONNECT,123456 789012<cr_lf>	N.A.
ATDHFTP	<cr_lf>OK<cr_lf><cr_lf>FTP DISCONNECT<cr_lf>	N.A.
FTP Directory		
ATFTPB	<cr_lf>OK<cr_lf><XML Directory Information> <cr_lf>BROWSE COMPLETE<cr_lf>	N.A.
ATFTPSUB	<cr_lf>OK<cr_lf>	N.A.
ATFTPUP	<cr_lf>OK<cr_lf>	N.A.
ATFTPROOT	<cr_lf>OK<cr_lf>	N.A.
FTP Push/Pull		
ATFTPPUSH	<cr_lf>OK<cr_lf><cr_lf>SENT,0001<cr_lf><cr_lf>SENT,0051 <cr_lf><cr_lf>PUSH COMPLETE<cr_lf>	N.A.
ATFTPPULL	<cr_lf>OK<cr_lf><cr_lf>File,0030,<30 Bytes of Data><cr_lf><cr_lf>PULL COMPLETE<cr_lf>	N.A.
OPP Con/Discon		
ATDOPP	<cr_lf>OK<cr_lf><cr_lf>OPP CONNECT,12345 6789012<cr_lf>	N.A.
ATDHOPP	<cr_lf>OK<cr_lf><cr_lf> OPP DISCONNECT<cr_lf>	N.A.
OPP Push/Pull		
ATSBCARD	<cr_lf>OK<cr_lf><vCard Data><<cr_lf>PUSH COMPLETE<cr_lf>	N.A.
ATRBCARD	<cr_lf>OK<cr_lf><cr_lf><vCard Data><cr_lf><cr_lf>PULL COMPLETE<cr_lf>	N.A.
ATOPPPUSH	<cr_lf>OK<cr_lf><cr_lf>PUSH COMPLETE<cr_lf>	N.A.
OPP Server		
ATBCARDR	<cr_lf>OPP CONNECT,123456789012<cr_lf> <cr_lf>PUSH START,<File Name>,<Size> <cr_lf><cr_lf><Data><cr_lf><cr_lf>PUSHCOMPLETE<cr_lf> <cr_lf>OPP DISCONNECT<cr_lf>	N.A.

Appendix F – Glossary, Acronyms, and Abbreviations

API – Application Protocol Interface
AT – Automatic Tone
ATMP – Attention Multi-Point
ASCII – American Standard Code for Information Interchange
BT – Bluetooth
BTW – Bluetooth Windows Stack
COD – Class of Device
COM – Communications
CR – Carriage Return
CTS – Clear To Send
DSR – Data Sent Receive
GND – Ground
HCI – Host Controller Interface
IP – Internet Protocol
LF – Line Feed
MCU – Microcontroller Unit
MISO – Master In Slave Out
MOSI – Master Out Slave In
NC – Not Connected
PC – Personal Computer
PCB – Printed Circuit Board
PCM – Pulse Code Modulation
PAN – Personnel Area Networking
PIN – Personal Identification Number
PIO – Pin Input/Output
RST – Reset
RTS – Ready to Send
RX – Receive
RST – Reset
RTS – Ready To Send
RX – Receive
SCO – Synchronous Connection-Oriented: the links used by BT to send voice.
SMT – Surface Mount Technology
SPI – Serial Protocol Interface
SPICK – SPI Clock
SPICS – SPI Chip Select
TTL – Transistor Transistor Logic
TX – Transmit
UART – Universal Asynchronous Receiver/Transmitter
USB – Universal Serial Bus
UUID – Universal Unique Identifier – maintained by Bluetooth SIG.
VCC – DC Power
VDD – DC Power
VM – Virtual Machine

Index

+

+++ – Put Device into Command Mode 37

A

ASCII bytes 6
AT Prefix +++ 13

B

BCARDR – Handle Business Card Request 55
Business Card Pull – RBCARD 54
Business Card Push – SBCARD 54
Byte Gaps 37

C

Cancel - UCL 38
Class of Device - SC 23
Clear All Paired Devices - CPAIR 39
Clear Sniff - CSNIFF 41
Command Mode 37
Configuration Parameters 6
CPAIR – Clear All Paired Devices 39
CSNIFF – Clear Sniff 41

D

Data Latency 37
Data Mode 37
DC – Dial Channel 33
DFTP – Dial FTP 48
DH – Dial Hang Up 36
DHC – Dial Hang Up by Channel 36
DHFTP – Dial Hang Up FTP 49
DHOPP – Dial Hang Up OPP 53
DHSCO – Dial Hang Up SCO 44
DI - Inquire 31
Dial as Master - DM 32
Dial as Slave - DS 36
Dial Channel - DC 33
Dial FTP – DFTP 48
Dial Hang Up - DH 36
Dial Hang Up by Channel - DHC 36
Dial Hang Up FTP – DHFTP 49
Dial Hang Up OPP – DHOPP 53
Dial Hang Up SCO - DHSCO 44
Dial Last - DL 34
Dial OPP – DOPP 53
Dial SCO - SCO 44
DL – Dial Last 34
DM – Dial as Master 32
DOPP – Dial OPP 53
DS - Dial as Slave 36
DSCO – Dial SCO 44

E

Enable Auto Sniff - SSNIFF 41
Enable Park - PARK 42
Enable Sniff - SNIFF 40
Example - Client Applications 61
Example - Master Discovery / Connection Sequence 59
Example - Slave Command Sequence 60
Examples - Server Applications 60, 61
Exit Park - XPARK 42
Exit Sniff Mode - XSNIFF 41

F

Factory Default Settings 58
Factory Reset - FRST 14
Fast data mode 37
FRST – Factory Reset 14
FTP Browse – FTPB 49
FTP client 48
FTP Pull – FTPPULL 51
FTP Push – FTPPUSH 51
FTP Server Example Using Broadcomm Stack 52
FTP Server Push Example Using Broadcomm Stack 52
FTPB – FTP Browse 49
FTPPULL – FTP Pull 51
FTPPUSH – FTP Push 51
FTPROOT – Move to FTP Root Directory 50
FTPSUB – Move to FTP Sub Directory 50
FTPUP – Move to FTP Parent Directory 50

G

Get BT Address 15
Get Channel UUIDS 18
Get Class of Device COD 16
Get Connection, Comm, Unconnected UART, Default Service Modes 16
Get Current Connection Status 15
Get Deep Sleep Mode 17
Get Escape Character 17
Get Friendly Name 15
Get FTP/OPP Connection Status 18
Get Inquiry & Master Connect Timeout 17
Get Link Quality - LQ 43
Get Link Quality by Channel - LQC 43
Get Link Supervisory Timeout 18
Get List of Paired Addresses 18
Get master Auto-Connect Address 16
Get Max TX Power Level 17
Get PIN Lock Mode 17
Get PIO5 Pulse Rate 17
Get Response, Security, AutoSCO, Filter Settings 16
Get RSSI Value - RSSI 42
Get RSSI Value by Channel - RSSIC 43
Get Security Level – RSW 2 19

Get Service Name 15
Get Slave Scan Intervals and Windows 16
Get Sniff Settings 18
Get Specific Transmission Channel 18
Get UART Settings 16
Glossary, Acronyms, and Abbreviations 85
Golden Receive Power Range 42

H

Handle Business Card Request – BCARDR 55
HyperTerminal 6

I

I/O pins 13
IL – Last Inquiry 32
Inpair by Index - UPAIR 39
Inquire – DI 31

L

LAST – Read Last Address 34
Last Inquiry – IL 32
Link Supervisory Timeout - LSTO 45
Lock User Settings - SW26 29
LQ – Get Link Quality 43
LQC – Get Link Quality by Channel 43
LSTO – Link Supervisory Timeout 45

M

MACLR – Master Address Clear 35
Master Address Clear – MACLR 35
MD – Put Device into Data Mode 38
MF – Put Device inot Fast Data Mode 38
Move to FTP Parent Directory – FTPUP 50
Move to FTP Root Directory – FTPROOT 50
Move to FTP Sub Directory – FTPSUB 50
Multi-Point (MP) Architecture 8
Multi-Point Example 62, 64

O

Object Push – OPPPUSH 55
OP - Overwrite PIN 23
OPP client 53
OPP Server Receive Business Card Example Using Broadcom Stack 56
OPP Server Send Business Card Example Using Broadcom Stack 56
OPP Server Send Data Example Using TransSend 57
OPPPUSH – Object Push 55
Overwrite PIN - OP 23

P

PAIR – Pair Devices 39
Pair Devices - PAIR 39
PARK – Enable Park 42
Power Consumption Class 1 80
PR – Respond to Manual PIN Request 23
Protocol V2.0 5
Put Device into Command Mode - +++ 37
Put Device into Data Mode - MD 38
Put Device into Fast Data Mode - MF 38

R

RBCARD – Business Card Pull 54
READ – Read Variable 45
Read Auto Connect Master on Data – RSW7 35
Read Bypass for Factory Config – RSW 0 14
Read Last Address - LAST 34
Read Max Connection Number – RSW 3 19
Read PIN Request Handling Mode - RSW 6 22
Read PIO Level – SR21 31
Read PIO Logic – RSW4 46
Read Remote Radio Name by BT Address - RRN 20
Read Remote Service Name - RRSN 21
Read Service Name - RSN 21
Read Service Name by Channel - RSNC 21
Read Variable - READ 45
Repeater Example 66
Repeater Mode 12
Reset - URST 14
Responde to manual PIN Request - PR 23
RRN – Read Remote Radio Name by BT Address 20
RRSN – Read Remote Service Name 21
RSN – Read Service Name 21
RSNC – Read Service Name by Channel 21
RSSI – Get RSSI Value 42
RSSIC – Get RSSI Value by Channel 43
RSW 0 – Read Bypass for Factory Config 14
RSW 2 – Get Security Level 19
RSW 3 – Read Max Connection Number 19
RSW 6 – Read PIN Request Handling Mode 22
RSW4 – Read PIO Logic 46
RSW7 – Read Auto Connect Master on Data 35

S

SBCARD – Business Card Push 54
SC - Class of Device 23
SDIS – Set Discoverable 47
Select Transmission Channel – SWC 46
SESC – Set Escape Character 37
Set Master Default Address - SMA 35
Set Auto Connect Master on Data – SSW7 35
Set Bypass for Factory Config – SSW 0 14
Set Discoverable – SDIS 47
Set Escape Character - SESC 37
Set Max Number of BT Connections – SSW 3 19
Set Max TX Power Level - SPF 44
Set PIN - SP 22
Set PIN Request Handling Mode - SSW 6 22
Set PIO Logic – SSW4 46
Set Security Level – SSW 2 19
Set Service Name - SSN 20
Set Service Name by Channel - SSNC 21
Set SocketWireless Name - SN 20
Set Undiscoverable – UDIS 47
Short Response Mode Structure 72
SI – Status Information 15
SMA - Set Master Default Address 35
SN - Set SocketWireless Name 20
SNIFF – Enable Sniff 40
Sniff and Low Power Modes 40

SP – Set PIN 22
SPF – Set Max TX Power Level 44
SR21 - Read PIO Level 31
SSN - Set Service Name 20
SSNC - Set Service Name by Channel 21
SSNIFF – Enable Auto Sniff 41
SSW 0 – Set Bypass for Factory Config 14
SSW 2 – Set Security Level 19
SSW 3 – Set Max Number of BT Connections 19
SSW 6 – Set PIN Request Handling Mode 22
SSW4 – Set PIO Logic 46
SSW7 – Set Auto Connect Master on Data 35
Status Information - SI 15
STORE – Store Variable 45
Store Variable - STORE 45
SW20 – Write UART Settings 24
SW21 – Write Slave Scan Intervals and Windows 25
SW22 – Write PIO Direction 26
SW23 – Write PIO Level 27
SW24 – Write Response, Security, Auto SCO, Filter Settings 27
SW25 – Write Connection, Comm, Unconnected UART, Default Services Modes 28
SW26 - Lock User Settings 29
SW27 – Write LED Rate} 29
SW28 – Write Inquiry and Master Timeout Settings} 29
SW29– Write PIN Lock Mode} 30
SW30 – Write Deep Sleep Mode} 30
SWC – Select Transmission Channel 46

U

UCL - Cancel 38
UDIS – Set Undiscoverable 47
Universal Unique Identifiers (UUIDs) 79
Unpair by Bluetooth Address - UPAIRB 39
UPAIR – Unpair by Index 39
UPAIRB – Unpair by Bluetooth Address 39
URST - Reset 14
UUIDs 79

V

VER – Version Command 13
Version Command – VER 13

W

Write Deep Sleep Mode – SW30 30
Write Inquiry and Master Timeout Settings – SW28 29
Write LED Rate – SW27 29
Write PIN Lock Mode - SW29 30
Write PIO Direction – SW22 26
Write PIO Level – SW22 27
Write Response, Security, Auto SCO, Filter Settings – SW24 27
Write Slave Scan Intervals and Windows – SW21 25
Write UART Settings – SW20 24
Write, Connection, Comm, Unconnected UART, Default Services Modes – SW25 28

X

XPARK – Exit Park 42
XSNIFF – Exit Sniff Mode 41